

PROJECTS THEORY APPLICATIONS CIRCUITS TECHNOLOGY

NUTS
AND
VOLTS

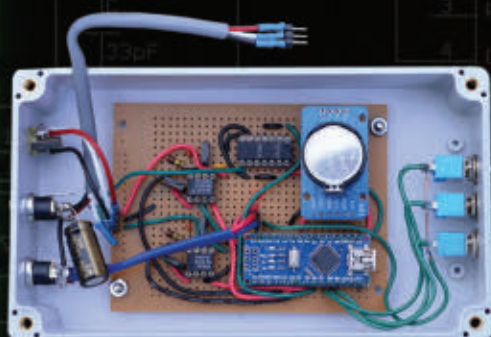
NUTS AND VOLTS

www.nutsvolts.com
November 2016

EVERYTHING FOR ELECTRONICS

Build the

JUMBO FULL-COLOR LED Matrix Display



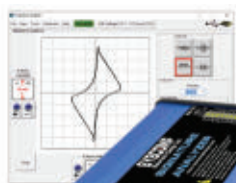
DIY 3-IN-1 TESTER
for your workbench

WI-FI WEATHER CLOCK
using the ESP8266

WORKING WITH STRINGS
in propeller spin

SHIELDING AND CABLES
what you need to know

More Unique Products



Signature Analyzer

- Controlled via USB
- 5-in-1 Functions
- Includes Oscilloscope AWC, I/O
- WIN, MacOS, Linux
- Under \$400

SIG-101

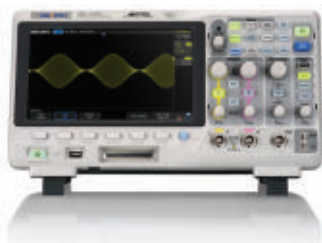
Oscilloscope, Waveform Generator, Digital I/O Port - All in One!



- Terrific educational tool with lessons/experiments
- AWC, I/O, PWM, FFT
- \$99.95

CGM-101

SDS1102X Scope Free Serial Decode Option



- 1GSa/s Sampling
- 14 Mpts Memory
- FFT
- 8" LCD
- 60,000 wfms/s
- \$424

Wide Selection of Spectrum Analyzers



- Desktop to USB stick versions cover up to 12GHz to find signal qualities or interference
- Starting at \$518

Power Supplies - 30W to 1500W



- Desktop to rack-mount power supplies to fit almost any need.
- Programmable on/off voltage level cycles
- From \$56

Automotive Diagnostic Equipment



- Picoscope test kits are designed to make diagnosing & testing vehicle electronics, components easy. Find faults fast!

EMC Precompliance Testing



- Check your PCB design before expensive formal testing to make sure you'll pass EMC specs with our affordable solutions.

PCB Repair Equipment



- Repair, don't throw away costly PCBs. BoardMaster systems are easy-to-use and can be used by anyone to find and repair PCB faults.

PRINTED CIRCUIT BOARDS

THINK YOU CAN FIND PCB PRICES THAT BEAT OURS?

WE DARE YOU.



THERE ARE NO GAMES INVOLVED IN OUR PRICING



Our Capabilities:

- From same day quick turn prototype to production in under 10 days
- Full CAD and CAM review plus design rule check on ALL Gerber files
- Materials: Fr4, Rigid, Flex, Metal Core (Aluminum), Polyimide, Rogers, Isola, etc.
- HDI Capabilities: Blind/Buried Microvias, 10+N+10, Via-in-Pad Technology, Sequential Lamination, Any Layer, etc.
- Our HDI Advantage: Direct Laser Drilling, Plasma De-Smear Technology, Laser Microvia, Conductive Plate Shut.

Take the Accutrace Challenge and see WHY OUR PRICING CANNOT BE BEATEN

Accutrace inc.

www.PCB4u.com sales@PCB4u.com

● SAM & ITAR Registered ● I SO 9001 - 2008, ● UL E33047

22 ESP8266 Weather Clock

If you recall my NTP clock utilizing the ESP8266 module, then you'll appreciate this "upgrade" to it that combines a weather station and timepiece all in one.

■ By Craig A. Lindley

28 A 3-in-1 Tester for Your Lab

This article describes a piece of test equipment that combines a variable DC voltage reference, a programmable function wave generator, and a fully functional frequency counter, and it costs less than \$150.

■ By Jim Teixeira

36 Using Programmable RGB LED Strips in Arrays

I've always wanted to build an LED table, and using LED strips made it super easy. Here's how I did it.

■ By Theron Wierenga



Page 28

Departments

06	DEVELOPING PERSPECTIVES	44	NV WEBSTORE
	<i>Superfluous Technology</i>	53	CLASSIFIEDS
07	READER FEEDBACK	62	TECH FORUM
20	NEW PRODUCTS	65	ELECTRO-NET
21	SHOWCASE	65	AD INDEX

Columns

08 TechKnowledgey 2016

Events, Advances, and News

Get the latest info on a supremely dense memory that's been created, the world's smallest reed switch, edible batteries, tubes with Bluetooth, plus other techie stuff.

12 The Ham's Wireless Workbench

Practical Technology from the Ham World

Shielding and Shielded Cables

For low frequency control, switching, and DC power, it is possible to get the job done without worrying too much about wiring and cabling practices. However, when RF gremlins begin to appear, suddenly, a whole new set of cautions and constraints gets piled on to your "simple" project. This column provides an overview of shielding, including some practices you should know as a defense against these gremlins.

16 Q&A

Reader Questions Answered Here

Repurposing X10 cameras and the real reason op-amps are called what they are head up this month's discussion.



46 The Spin Zone

Adventures in Propeller Programming

Strings & Things

A frequent topic of discussion by newcomers on the Propeller forum is the use of strings. Spin doesn't have an explicit string type, so this can be a little confusing at first. Have no fear! Working with strings is very simple.

54 The Design Cycle

Advanced Techniques for Design Engineers

Nordic's New nRF52832 Gets You On Track with BLE Apps

BLE is a BIG thing these days. Get in on how to design the hardware and write the code for Nordic Semiconductor's ubiquitous nRF5 SoCs. In this series of discussions, we will utilize Nordic's BLE development tools and the Keil Cortex-M C compiler to bring a BLE project to life. Follow along as Fred takes you through the complete Nordic BLE device design cycle.

2-For-1 Holiday Offer



Give a one year (12 issues) subscription to a friend and receive a *FREE* one year subscription for another friend, or yourself!

Not only will they get a full year of Nuts & Volts delivered right to their door, they'll also get free access to our digital (web) edition, PLUS our mobile (iOS, Android, Kindle Fire) edition.

Subscribe online at:
www.nutsvolts.com

or call:

877-525-2539 (toll free)
818-487-4545 (outside US)

Be sure to use promotion code
Y6WXMSA when ordering

U.S. Price: 1 Yr - \$26.95

DEVELOPING PERSPECTIVES

by
Bryan
Bergeron,
Editor

Superfluous Technology: Avoiding the Binge

This is a great time to be an electronics enthusiast. Electronics are relatively inexpensive, readily available, and focused on just about every aspect of life — from home to work to play to health. However, in this time of technology abundance, it's also easy to over-indulge in what you want versus what you may need.

I was reminded of this fact in the process of upgrading my sound system. Years ago, when I installed my previous sound system, I added RF chokes to the power and signal lines of my amplifier, receiver, preamp, and speakers. I had RF-proofed everything as a carryover habit from my early ham radio days, when direct and rectified signals from my homebrew CW rigs had a habit of making their way into AM and FM radio as well as TV signals. In fact, the chokes were an unnecessary expense and didn't add to the signal/noise ratio of my previous sound system. My new cables are standard off-the-shelf cords, and the sound is superb.

With that realization, I started exploring my home and office for additional signs of over-indulgence in technology. In my spring cleaning effort, I came up with a box full of technology that I should have never purchased, ranging from cable adapter kits that I bought "just in case" to fitness tracker devices that have maybe a couple hours of use. Sure, everything I tossed into the box was a deal when I bought it, but I clearly didn't need it. I would have been better off saving my money and either putting it toward a better bench top DMM or other gear that I use on a regular basis, or expanding my investment portfolio.

Hindsight is, of course, 20-20, and I'd like to think that I'm a more thoughtful consumer of technology today. Still, when I start exploring a new area of technology or revisit one that's changed significantly in the past few years, it's hard for me to know what's necessary and what's a waste of money and time. For example, when I first started working with drones, I ordered several multi-function testers for the battery packs, electronic speed control (ESC) settings, and motor performance. Turns out that all I needed was a simple meter to tune the motors, and a smart charger to monitor the batteries. At the time, I didn't want to get caught without a critical piece of test gear and then have to wait weeks for delivery, so I spent (i.e., wasted) the extra money.

I narrowly avoided a technology binge when I upgraded my smart home system to add remote video monitoring to an installed base of remote lighting and temperature control technology. Still, I was torn between erring on the side of too much technology rather than purchase a system without capabilities that I might want in the future. It helped that in selecting remote lighting, I went through several generations of smart lights before I found a technology that exactly fit my needs. Although I enjoyed the process, I felt guilty wasting perfectly good LED bulbs with decades of life remaining. I wasn't going to repeat the mistake with a smart alarm system.

My current challenge is with smart home appliances. Take a refrigerator. Beyond keeping food at a constant low temperature, current

NUTS AND VOLTS

Published Monthly By
T & L Publications, Inc.

430 Princeland Ct.
Corona, CA 92879-1300
(951) 371-8497

FAX **(951) 371-3052**

Webstore orders only **1-800-783-4624**

www.nutsvolts.com

Subscription Orders

Toll Free **1-877-525-2539**

Outside US **1-818-487-4545**

P.O. Box 15277

North Hollywood, CA 91615

FOUNDER

Jack Lemieux

PUBLISHER

Larry Lemieux

publisher@nutsvolts.com

ASSOCIATE PUBLISHER/ ADVERTISING SALES

Robin Lemieux

robin@nutsvolts.com

EDITOR

Bryan Bergeron

techedit-nutsvolts@yahoo.com

VP OF OPERATIONS

Vern Graner

vern@nutsvolts.com

CONTRIBUTING EDITORS

Fred Eady

Kristen McIntyre

Jeff Eckert

Jim Teixeira

Craig Lindley

Jon McPhalen

Ward Silver

Theron Wierenga

CIRCULATION DEPARTMENT

subscribe@nutsvolts.com

SHOW COORDINATOR

Audrey Lemieux

WEBSTORE MARKETING COVER GRAPHICS

Brian Kirkpatrick

sales@nutsvolts.com

WEBSTORE MANAGER/ PRODUCTION

Sean Lemieux

sean@nutsvolts.com

ADMINISTRATIVE STAFF

Re Gandara

Copyright © 2016 by T & L Publications, Inc.

All Rights Reserved

All advertising is subject to publisher's approval. We are not responsible for mistakes, misprints, or typographical errors. *Nuts & Volts Magazine* assumes no responsibility for the availability or condition of advertised items or for the honesty of the advertiser. The publisher makes no claims for the legality of any item advertised in *Nuts & Volts*. This is the sole responsibility of the advertiser. Advertisers and their agencies agree to indemnify and protect the publisher from any and all claims, action, or expense arising from advertising placed in *Nuts & Volts*. Please send all editorial correspondence, UPS, overnight mail, and artwork to: 430 Princeland Court, Corona, CA 92879.

models offer features ranging from built-in PDAs to at least one model that takes a digital image of the refrigerator contents to remind you of what you need to purchase at the grocery store. Clearly, I don't need a dedicated refrigerator camera system.

Perhaps I'll revisit smart refrigerators when there's an

affordable model that features image recognition and can automatically generate a list of what I need to pick up at the store — or have the refrigerator email the local grocery store for a delivery. Of course, I don't really *need* this sort of technology today, but perhaps in 25 years or so when I'm less mentally and physically

able I'll appreciate not having to think about a grocery list.

All of this said, most electronics enthusiasts identify with being an early adopter of technology. And there's nothing wrong with that — as long as you act responsibly and remain aware of your reasons for buying a particular technology. **NV**

READER FEEDBACK

Couldn't Resist

In the August 2016 issue on page 24 at the end of the first full paragraph, it says "... any voltage less than 2,000 volts will be safely ..." The Parts List on page 25 doesn't specify any power rating (and I didn't dig through Digi-Key's catalog for R1's rating), but a look on page 26 in Figures 7 and 8 doesn't show any RN 70s. Typical 1/4W resistors are rated 250 VDC and 1/2W 350 VDC with over-voltage perhaps double for some limited time duration on the OV. (Modern resistors of higher ratings don't necessarily scale up for some unknown [to me] reason.)

The first 200V 'division' should be no problem, but the second is iffy, and any higher level inputs are definite overloads. I don't know what the character of the let through current is on breakdown, so I don't know how well the op-amp could handle it.

Over the years, I've noticed a number of circuits — particularly

metering — where the voltage ratings of the input scaling resistors are ignored, and I had to finally say something.

I was particularly drawn to this article as I have a Velleman EDU08 'scope that has a 30V peak input limit which, in general, is fine for digital work, but is limited for audio power amps and other 'archaic' electronics; hollow state in particular.

As a corollary to the RESISTOR voltage problem, please check the voltage rating on the test leads!

Yes, I go back a few years to Ziff Davis and Hugo Gernsback publications. I started reading NV when it was more like a newsletter!

Ron G.

The type of resistor used may result in break-over conduction, however, the .4 inch lead spacing is sufficient to prevent arc-over conduction. The resistor used may have internal break-over ratings depending on the ceramic coating,

the metalized conductor spacing, and the type of ceramic core material. If the resistor does not have high enough ratings, break-over could occur and result in damage to the IC. Be sure the one meg and nine meg resistors have sufficient ratings. Another method would be to put lower value resistors in series to add up to the needed values to avoid the full voltage drop across an individual resistor.

It just goes to show how astute and informed Nuts & Volts readers are.

Ron Hoffman

I used to have a supply of "HV" resistors from then-new CRT TV sets. Get for discharging HV from vacuum tubes, etc. As far as the current crop of leaded resistors go, I hear you. I'm not sure why the breakdown voltage isn't scalable.

Thanks for sharing your thoughts with the readers.

Bryan Bergeron, Editor

Comments from the "Join the Conversation" question listed below that was featured in one of Nuts & Volts weekly content newsletters:

The electronics hobby is always evolving and hobbyists are always learning. What book, program, mentor, or other source of information has helped you the most to get to where you are? Are you self-taught or do you have formal electronics training?

I started in the late '40s by taking apart my toys I got for Christmas and birthdays. Some had batteries. It all began before television. I built a crystal set and put up an antenna on the roof. Then, I got some kits like

one tube and one, two, and five transistor radios. I discovered Allied Radio and bought more kits and some small reference books. I studied the specs of many different kits, buying several from Knight, Heath,

and Eico. I learned Ohm's law and how to solder along the way. I got to the point of building audio amplifiers using 20 watt 6L6 push-pull output.

I built some bass reflex speaker cabinets. I just barely got into solid-state as my energies went into automotive and girls. I am still very much interested in the vintage electronics and setting up my work shop to continue with the early solid-state electronics.

Robert Holzman

Continued on page 61

ADVANCED TECHNOLOGY

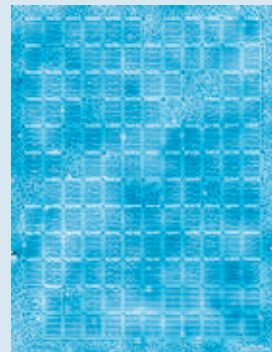
Supremely Dense Memory Created

Every day, the world creates in excess of a billion gigabytes of new data, and people need somewhere to store it. Even with today's increasingly dense storage media, that still takes up a lot of physical space. As early as 1959, legendary physicist, Richard Feynman surmised that if we had a way of arranging a batch of individual atoms in a precise orderly pattern, we could use them to store information with the ultimate density: one bit per atom. As usual, Feynman was correct, and a scientific team at the Kavli Institute of Nanoscience at the Netherlands' Delft University (kavli.tudelft.nl) — inspired by Feynman's famous lecture, "There's Plenty of Room at the Bottom," — recently reported accomplishing the feat.

Lead scientist, Sander Otte, in a paper published in *Nature Nanotechnology*, described building a 1 kB memory in which each bit is represented by the position of one chlorine atom, thus achieving a storage density of 500 Tb/in² — 500 times better than the best available commercial hard drive. "In theory," he noted, "this storage density would allow all books ever created by humans to be written on a single post stamp."

■ Atomic-level memory containing part of a Richard Feynman lecture (with text added).

Shown here is a scanning tunneling microscope scan of the 96 x 126 nm (about 0.000004 x 0.000005 inch) memory, in which a portion of the lecture is stored (text markup added). Each atom (perched on the copper underlayer) can occupy either of two positions. In the top position, there is a hole beneath it, and this represents a one. If the atom is pushed to the bottom position, it becomes a zero.



Obviously, a 1 kB memory isn't particularly useful, but scaling it up isn't the main obstacle to creating a commercial version. Otte also revealed, "In its current form, the memory can operate only in very clean vacuum conditions and at liquid nitrogen temperature (77 K [-321°F]), so the actual storage of data on an atomic scale is still some way off. But through this achievement, we have certainly come a big step closer." ▲

Edible Batteries

By now, we've all heard about ingestible cameras and other diagnostic devices designed to be swallowed, perform specific functions within the digestive system, and eventually exit via excretion. For one-time or even occasional use, there is little risk of such equipment getting stuck in the gastrointestinal tract and causing problems. However, the risk increases significantly with multiple or periodic use, and sooner or later, these devices are bound to cause problems for someone. Because these devices are powered by batteries that contain toxic substances, there is some risk of poisoning the patient. This led Dr. Christopher Bettinger, of the Materials Science and Engineering Dept. at Carnegie Mellon University (www.materials.cmu.edu), to wonder if it would be possible to develop nontoxic edible batteries.

"For decades," he observed, "people have been envisioning that one day, we would have edible electronic devices to diagnose or treat disease. But if you want to take a device every day, you have to think about toxicity issues. That's when we have to think about biologically derived materials that could replace some of these things you might find in a RadioShack."

Bettinger's team decided to take a look at melanins and other naturally occurring compounds. In our bodies,

melanins absorb UV light to snuff out free radicals and protect us from damage. They also happen to bind and unbind metallic ions. "We thought, this is basically a battery," Bettinger noted. The researchers experimented with battery designs that use melanin pigments at either the positive or negative terminals. They experimented with a variety of electrode materials including manganese oxide, and sodium titanium phosphate, and cations such as copper and iron that the body already normally uses.

"We found basically that they work," said Hang-Ah Park, Ph.D., a post-doctoral researcher at CMU. "The exact numbers depend on the configuration, but as an example, we can power a 5 mW device for up to 18 hours using 600 milligrams of active melanin material as a cathode."

Because ingestible devices aren't intended to be in the body for more than 20 hours or so, that is just fine. In the future, Bettinger envisions using such batteries for sensing gut microbiome changes, and responding with a release of medicine or for delivering bursts of a vaccine over a period of several hours. ▲



■ Dr. Christopher Bettinger shows off his edible nontoxic battery.

COMPUTERS and NETWORKING

Notebook Aims at Small to Medium Businesses

According to ABI Research and other research organizations, while the PC market rapidly evolves away from desktop computers, laptop sales are remaining fairly steady. ABI predicts that sales of ultraportable models (i.e., machines that typically weigh less than 4 lb and are no more than 1.5 in thick when closed) will shoot up from 27.1 million last year to 30 million this year, and 41.2 million by 2021. Aimed squarely at that market is Acer's new TravelMate X3 lineup, the first of which is the X349 — slated to hit the market by the time you read this.

The X3 notebooks are designed to appeal to professionals who work in small and medium size businesses. The X349 — built on an all-aluminum chassis — is less than 18 mm (0.7 in) thick, weighs only 1.53 kg (3.3 lb), and provides up to 10 hr of battery life. Inside the box you get a sixth-generation Intel Core processor (unspecified as of this writing), up to 8 GB of DDR4 memory, and up to 512 GB of SSD storage. It

■ Acer's new TravelMate X349 notebook.



comes with Windows 10 Pro and an integrated touch fingerprint sensor for security. "Fine-tuned" speakers and microphones are built in, along with a 720p webcam and a 14 in HD IPS display. Optional is a USB type-C dock which allows connection of up to two 4K displays.

The base price is \$649.99. According to Acer, exact specs, prices, and availability vary by region, so visit www.acer.com if you want details. ▲

No Kidding, a 60 TB SSD

It probably will be of only theoretical interest unless you operate a high-level enterprise data center, but it is still worth noting the latest Flash memory innovations from Seagate (www.seagate.com). Most impressive is a serial attached SCSI solid-state drive with a capacity of 60 TB. Yes, that's 60 terabytes — the highest capacity SSD ever demonstrated. That's enough to store 400 million typical photos or 12,000 DVD movies. The unit is currently just a demo technology, so it doesn't even have a name yet. However, Seagate says a commercial version should be available sometime in 2017. Just pick up 17 of them, and you'll have a full petabyte of available storage.

A little more down to earth is the company's 8 TB Nytro® XP7200 NVMe SSD. Like the 60 TB version, it is designed to accommodate the hyperscale needs of data centers that need to quickly access and process huge amounts of data such as in scientific research and weather modeling operations. It features a single PCIe interface and four separate controllers that boost the processing power up to four times faster than single-controller drives.

No prices were quoted for either drive, but "cheap" isn't a term that's likely to be applicable in either case. Just for comparison, we note that Samsung's 15.36 TB unit went on sale a few months ago with a price tag just north of \$10,000. ▲



■ Seagate will begin selling a 60 TB SSD next year.

Memoirs of a Pioneer, Free

■ Gary Kildall, PC innovator.

Nearly everyone is familiar with Bill Gates, but you may never have heard of Gary Kildall. Strangely enough, it could easily have been the opposite. Way back in 1973, Kildall developed the first high-level microprocessor programming language called PL/M, for *Programming Language for Microcomputers*. The same year, he came up with CP/M (*Control Program for Microcomputers*) — the first PC disk operating system. By 1981, it was the de facto industry standard OS, running on 3,000 different computers, and Kildall's company, Digital Research, Inc. (DRI), had yearly revenues of about \$5.5 million. About that time, IBM decided to get into the PC business but elected not to develop its own OS. The company approached Kildall with the idea of buying a future version (called CP/M-86) for the upcoming IBM PC. History is a bit hazy as to why the negotiations fell through, but suffice it to say that Gates was happy to step in and become the billionaire that Kildall could have been. (You needn't shed too many tears for him, though, as he sold DRI to Novell in 1991 for close to \$80 million worth of stock.)

In any event, Kildall died in 1994 at the age of 52. A year before, however, he wrote the draft of a memoir called, *Computer Connections: People, Places, and Events in the Evolution of the Personal Computer*



Industry. Earlier this year, his children decided to make a substantial portion of the manuscript available to the public, and it's now a free download available at the Computer History Museum. Those who are interested in the deep dark annals of PC history need only visit www.computerhistory.org to indulge. ▲

CIRCUITS and DEVICES

World's Smallest Reed Switch

The reed switch is not particularly new or high tech, having been invented in 1936 by Bell Labs' W. B. Ellwood. It is a handy little component, though, that is useful in applications ranging from simple position sensing to RF switching, robotics and automation, marine applications, and so on. It is therefore of interest that Standex Electronics has introduced the GR150, billed as the smallest magnetic reed switch in the world, measuring only 3.7 mm (0.15 in) long.

Designed for use where the available magnetic field is very low or space is extremely limited, it is said to be particularly useful in medical devices such as hearing aids and pill cams, as well as in consumer electronics and industrial, military, and aerospace applications. Specs include a max power rating of 1 VA, max switching current of 0.055A DC, and max carry current rating of

0.5A DC. Sensitivity ranges from 3 to 20 ampere turns with an operating temp range of -40°C to +125°C. Details are available at www.standexelectronics.com/gr150. ▲



■ The Standex GR150 ultraminiature reed switch.

Tubes with Bluetooth

For those of us who grew up in the era of vacuum tubes and vinyl, today's digital music tends to sound somewhat cold and flat. On the other hand, if you hit the Salvation Army and pick up an old Dynaco tube receiver, you'll be looking at about 1% THD and a complete lack of modern accoutrements such as Bluetooth. Lo and behold, there is a solution in the form of the Monoprice Stereo Hybrid Tube Amp with Bluetooth, available at www.monoprice.com (if you search hard enough) or via Amazon.

According to the dealer, "Because of its lossy nature, digital music can often lose some of the depth and warmth that gives it life. Additionally, wireless streaming adds an additional layer of digital compression, which can further degrade the audio quality. The addition of the tube preamplifier stage adds back some of the warmth and richness, breathing new life into your digital music."

Specs include 25W/channel peak (15W RMS), less than 0.1% THD, Bluetooth v4.0 wireless streaming, and a solid 20 Hz to 20 kHz frequency response. The 85 dB SNR isn't ideal, but good enough for most ears. The amp also

features an analog stereo input via RCA jacks, five-way binding post speaker connections, and a modest \$150 price tag. Best of all, the exposed tubes look super cool when they're warmed up and running. ▲

■ Monoprice's \$150 stereo amp features vacuum tubes plus wireless streaming.



INDUSTRY and the PROFESSION

Video Games Turn 50

It seems odd, but this year marks the 50th anniversary of the video game console. The original unit — known as the "Brown Box" — was invented in 1966 by Rudolf Henry Baer, a German-born engineer who was working at Sanders Associates at the time. The device was the first interactive video game system to use a home TV set as the viewing screen, and a later version went into commercial production as the Magnavox Odyssey system.

In subsequent years, Baer helped develop other consoles and consumer games and, in 2006, was awarded the National Medal of Technology for "his groundbreaking and pioneering creation, development, and commercialization of interactive video games, which spawned related

uses, applications, and mega-industries in both the entertainment and education realm." He passed away in 2014 at the age of 92. NV

■ The Brown Box video game console, 1967 model.



Shielding and Shielded Cables

Rarely on the schematic, but crucial to good performance!

For low frequency control, switching, and DC power, it is possible to get the job done without worrying too much about wiring and cabling practices. If Point A is supposed to be connected to Point B and the ohmmeter says it is — that's generally sufficient. The "fun" usually begins when the operation of Circuit A starts disrupting the operation of Circuit B, or if the project involves low-level audio such as from a microphone or pickup coil. If there is a nearby transmitter of any sort or perhaps a sensitive receiver, RF gremlins might begin

Loops: Keep Them Small

Back in the early 19th century, Faraday discovered that a changing magnetic field enclosed by a wire would induce a voltage in that wire. This relationship became known as Faraday's Law. Any loop formed by conductors — whether they are wires in a piece of equipment, conductors in a cable, or traces on a printed circuit board (PCB) — will pick up a signal from a stray magnetic or electromagnetic field. Similarly, varying current flowing in a loop will create and radiate a signal that can be a source of interference.

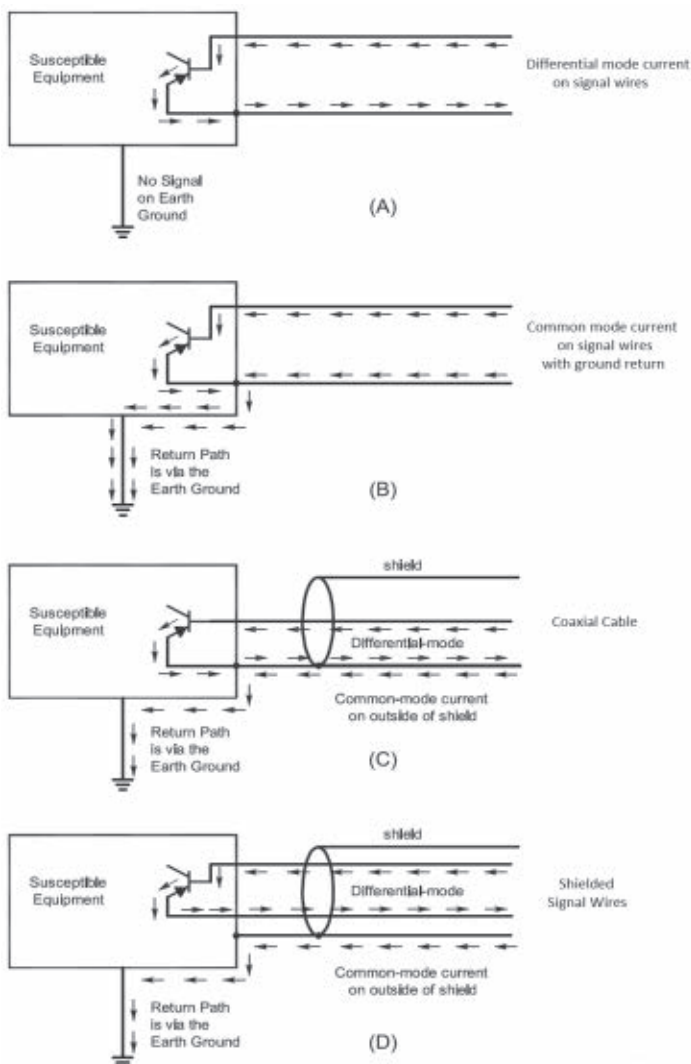
An easy and inexpensive way to keep a circuit from becoming a source or victim of interference is to simply twist the wires together so that the area of the loop they form is small. Or, lay out PCB traces for a signal and its return path next to each other. The higher the current carried by the loop or nearby conductors (such as for motors or appliances), the more important this becomes.

to appear. Suddenly, a whole new set of cautions and constraints gets piled on to your "simple" project. In this column, I'm going to provide an overview of shielding, including some practices you should know as a defense against these gremlins.

Shielded Cables

Most readers will have some sort of shielded cable in their parts inventory. Maybe it is a shielded audio cable with one or two conductors inside an outer shield of thin foil or fine wires wrapped around them. Maybe it's a multi-conductor control cable with a braided shield.

■ **FIGURE 1.** Examples of common mode and differential mode signals for different types of cables. A shows differential mode signal current in a two-wire unshielded cable. B shows a common mode signal current with a return path via the enclosure and the earth ground. In C, a common mode signal flows on the outside of a coaxial cable shield with a differential mode signal inside the cable. D shows the differential mode signal on signal wires inside a shield with common mode current flowing on the outside of the shield. (Graphic courtesy of the American Radio Relay League.)



Post comments on this article and find any associated files and/or downloads at
www.nutsvolts.com/magazine/article/November2016_HamsWirelessWorkbench-Shielding-Cables.

Either way, the goal of using these cables is twofold: to prevent (or at least reduce) signals inside the cable from coupling to other circuits or cables; and to prevent stray electromagnetic fields from creating voltages and currents in the conductors inside the shield. So far, so good.

In most of these cables — particularly those intended for use with audio or RF below 1 MHz — if you strip back the shield, you'll see that the conductors are twisted together. (The number of times per inch or cm that the wires are twisted together is called the pitch.) This is done for a couple of reasons. First, twisting the conductors keeps them pressed together so there is a minimum of area between them. This helps reduce the voltage induced in the conductors by an external field (see the **sidebar**).

Second, it maintains electrical balance of the conductors so they all look about the same to an external field. This keeps signals that may be coupled to the cable on all conductors (common mode signals) from being converted to signals that are not the same on different conductors (differential mode signals), and so interfere with the desired signals. **Figure 1** shows examples of common and differential mode signals on different types of cables.

So, doesn't the shield keep all fields out of the cable? Not really. At very low frequencies — such as AC power at 50/60 Hz — the magnetic fields from a transformer or power line go right through nearly all shields because the material is not thick enough. (See the **sidebar** on shielding effectiveness.) Even a high quality coaxial cable shield is not enough. Just feed some low-level audio through a piece of coax laying on a power transformer to hear the induced hum. As the frequency gets higher, the shielding becomes more effective, as long as it is connected properly.

The other common type of shielded cable is coaxial cable with a single center conductor surrounded by an outer conductor (shield), separated by a plastic or air dielectric. Current in coax flows on the outside of the center conductor and on the inside of the shield which is either a tightly woven braid of fine wires, a solid tube of metal (foil or thicker), or a combination of the two. Coax is very effective at shielding the signal flowing inside the cable from outside fields because of its excellent symmetry (which causes the effect of the fields to cancel) and because of the skin effect (see the **sidebar**) that isolates common mode current on the outside of the shield from the signal inside.

Shielded Enclosures

The higher the frequency of signals used by a circuit, the more important it is to use a metallic enclosure. Assuming wires and cables going in and out of the enclosure are properly routed and connected, the metal enclosure can keep RF from external fields out and RF from internal signals in. Furthermore, a metal enclosure provides

What is Skin Effect?

Because of how AC fields interact with excellent conductors, metals conduct AC only to a certain skin depth, d , that is inversely proportional to the square of the frequency. This is called the skin effect and it decreases the cross-section area of the conductor that conducts AC, increasing its resistance.

$$d = 1 / \sqrt{\pi f \mu s}$$

μ is the conductor's permeability and s is the conductor's conductivity.

The skin effect begins to have a significant effect above 1 MHz. By 10 MHz, a copper wire only conducts current in its outer 0.02 mm. At UHF and above, a thin layer of metal plating is sufficient.

mechanical strength to your project.

Shielding is important for sensitive or important circuits, and to be a "good neighbor" to other equipment that might be disrupted by radiated or conducted EMI (electromagnetic interference). As an example, if you have a general coverage receiver and live in a typical suburban or urban neighborhood, connect a dipole or other antenna and tune to 14.030 MHz — you will hear many weak "birdies" (single tones, often slightly wobbly or jumpy) from Ethernet network equipment being radiated by cables connected to unshielded or poorly shielded equipment. (Computer equipment is notorious for creating and responding to EMI into the UHF range.)

How Effective is Your Shield?

A magnetic field is attenuated by absorption as it passes through the thickness of a shield by 8.7 dB for each skin depth of thickness. (See the sidebar on skin depth.) Figure 2 is a graph of skin depth vs. frequency for common shielding materials. The smaller the skin depth for a shielding material, the greater the absorption it provides for a given thickness.

Only magnetic materials can provide magnetic shielding at AC power frequencies. Copper and aluminum thick enough to form a rigid chassis or enclosure begin to provide magnetic shielding around 10 kHz and are quite effective above 1 MHz.

Electric field shielding occurs by reflection, and the loss can be computed by:

$$R = 20 \log [(Z_w / 4Z_s) \cos \Phi] \text{ dB}$$

where R is the reflection loss, Z_w is the wave impedance (377Ω in free space), Z_s is the impedance of the shield, and Φ is the angle between the field and the shield.

For common shield materials like aluminum and copper, Z_s is $<1\Omega$ so even a very thin shield can provide more than 70 dB of shielding. Electric field shielding is quite easy — especially at low frequencies. An electric field shield must be continuous and must completely cover the circuit to be shielded.

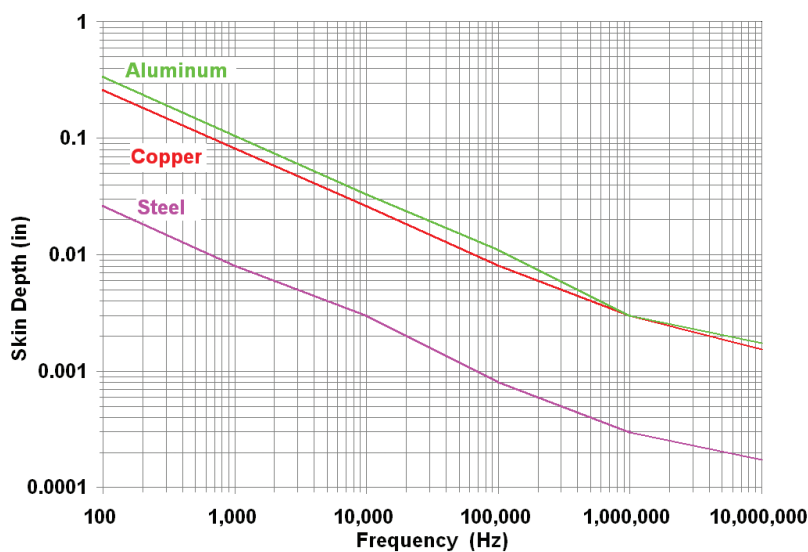
What's the Buzz?

There is a tendency to consider any unwanted low frequency tone in our electronics as “hum,” but more frequently the tone is really a “buzz.” True hum at 50 or 60 Hz is a sine wave created by a magnetic field from AC current or a power transformer. Buzz, on the other hand, sounds harsher with higher frequency components.

If the buzz has a fundamental frequency of 120 Hz, it is almost always caused by insufficiently filtered ripple from a full-wave rectifier in a power supply. If the buzz includes not only the 50/60 Hz tone but many higher harmonics, this is usually the result of improper AC neutral circuit connections or wiring that carries currents with significant harmonic energy known as triplen harmonics. From the tone of the unwanted hum or buzz, you can generally get an idea of its source.

Check out the **sidebar** that discusses the shielding effectiveness of various types and thicknesses of metal. **Figure 2** provides a graph of skin depth for three common metals with frequency. As you can see, for all but the thinnest sheet metal, aluminum enclosures are quite effective at shielding above a few kHz. Aluminum is easy to work with and strong.

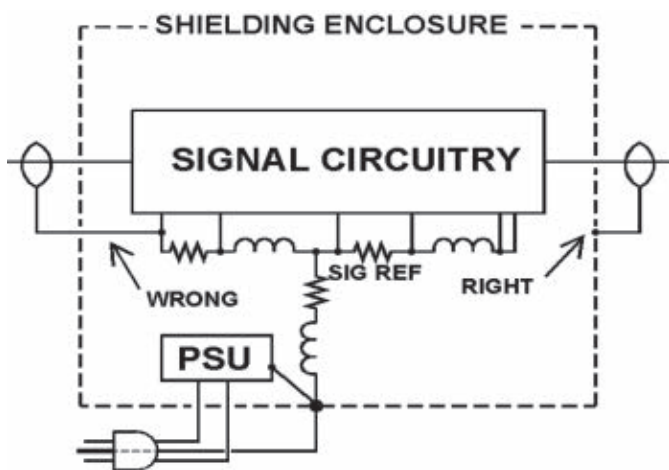
Other options include building your own enclosure out of printed circuit board material soldered together, bending up some brass or copper sheet, and even spraying plastic



■ **FIGURE 2.** Skin depth versus frequency for aluminum, copper, and steel.

enclosures with conductive paint.

Regardless of how good your enclosure is, its effectiveness can be ruined by mismanaging the connections of cable shields. **Figure 3** shows how a shield should be connected. Treat the shield as a sort of “RF water pipe” that keeps RF signals inside of the shield and



■ **FIGURE 3.** The correct and incorrect way of connecting a shield to an enclosure. At the left, bringing the shield through the enclosure allows any common mode current on the shield to couple into the circuit's signal reference (SIG REF) path. This is a very common source of RF interference both to and from the equipment. At the right, the shield is connected properly to the enclosure. (Figure courtesy of Jim Brown K9YC.)

Signals ala Mode

For regular zip cord feeding a speaker, the output signal from the audio amplifier causes current to flow out one conductor and in on the other. This is a differential mode signal in which there is a voltage between the conductors, and currents flow in opposite directions. An external field, however, would induce approximately equal currents or voltages on both conductors, causing current to flow on both conductors in the same direction. This is a common mode signal with current that flows equally on all conductors. (The current would presumably keep flowing from the speaker enclosure through any ground connection and back to the amplifier.)

If the common mode voltages and currents on each conductor are exactly equal, the net effect on equipment in the path of the common mode signal is very small. If there is even a slight imbalance between the conductors, however, that changes some of the common mode signal into a differential mode signal which can affect the equipment. Twisting unshielded zip cord helps maintain balance by insuring that both conductors are exposed to external fields in the same way. Coaxial cable can also be a host to common mode signals on the outer surface of its shield. Even though the signal may not be present on all conductors, it is also considered a common mode signal. These signals can be converted to differential mode signals if the shield connection allows common mode current to enter the inside of the coax. Proper shield connections are very important for equipment connected together with coax.

The “Pin 1 Problem”

In the professional audio industry, the problem of RF interference caused by improperly connected shields was originally identified in a 1994 paper by Neil Muncy. (See the end of this sidebar for the full reference.) This problem was named “The Pin 1 Problem” because pin 1 is the designated shield contact on the three-pin XL-style connectors widely used for microphones and line level audio signals. By demonstrating the nature of the problem — improper connections to the connector’s pin 1 — and how to fix it, “pin 1 problems” in professional audio equipment have been greatly reduced.

Hobbyists are often referred to professional audio and signal-processing literature on RFI that uses the term “pin 1 problem.” You should recognize this as referring to improperly connected shields that lead to RFI caused by common mode RF currents on the shields of connecting cables. The same cures that reduce susceptibility to RFI in audio equipment will work in radio, test, and control equipment too.

The professional audio literature and Audio Engineering Society (AES) standards are excellent sources of information on EMC, with amateurs playing a leading role in their development. Neil Muncy’s paper can be found as “Noise Susceptibility in Analog and Digital Signal Processing Systems,” presented at the 97th AES Convention of the Audio Engineering Society in San Francisco, CA, Nov. 1994.

keeps RF currents from external fields out of the enclosure.

NEVER bring a shield through a shielded enclosure. A shield should always be connected to the outside of a metal enclosure. By bringing the shield through the wall of the enclosure, you create a terrific pathway for RF to get in and out. And it will!

NEVER use the shield as a return or ground connection for a signal if you can avoid it. Any common mode current on the outside of the shield will follow the return path to the circuit ground and can wreak havoc. Similarly, any RF noise on the circuit ground will happily escape the enclosure by this same path and radiate to cause RFI.

A lot of audio equipment uses a shielded twisted-pair cable for left/right/common. What should you do if the equipment experiences RFI from external signals? Assuming your equipment has a metal enclosure, use shielded (metal) connectors and be sure there is a good connection between the cable shield and the connector’s back shell.



If the RFI persists, chances are good that one or two cables are providing the path for RF to get into the equipment. Remove all disconnectable cables and verify the RFI has stopped. Reconnect the cables one at a time until the RFI returns. Add ferrite cores (described in the July 2015 and September 2015 Ham’s Wireless Workbench columns) to the cable at the equipment experiencing the RFI. Use a type 31 mix for HF signals and a 43 mix for

upper HF and VHF/UHF signals. (Most inexpensive ferrite cores and beads are type 43, but ask the vendor to be sure.) Once you’ve cleared up the RFI from that cable, connect more cables one at a time to see if they need the ferrite treatment too.

References

I’ve used shielding to scratch only the surface of electromagnetic compatibility (EMC) — a subject that covers all sorts of interference, transients, and noise. If you would like more information on how hams view these and related problems, take a look through the free publications of Jim Brown K9YC. (**Figure 3** is from Jim’s tutorial, “A Ham’s Guide to RFI, Ferrites, Baluns, and Audio Interfacing.”) A retired audio engineer and active ham, he has written many papers on dealing with interference. You can find them at www.audiosystemsgroup.com/publish.htm.

Take a look in the “Hum, Buzz, and Interference” section. The ARRL’s tech portal (www.arrl.org/tech-portal) is another great source of know-how to deal with unwanted RF — incoming or outbound! **NV**




TS-7970

Industrial High Performance i.MX6
Computer with Wireless Connectivity
and Dual GbEth

- 1 GHz Single or Quad Core Cortex A9 ARM CPU
- 2 GB DDR3 RAM (Quad) and 4 GB MLC eMMC Flash
- Bluetooth 4.0+EDR and WiFi 802.11BGN onboard radios

\$169
QTY 100

www.embeddedARM.com



In this column, Kristen answers questions about all aspects of electronics, including computer hardware, software, circuits, electronic theory, troubleshooting, and anything else of interest to the hobbyist. Feel free to participate with your questions, comments, or suggestions. **Send all questions and comments to: Q&A@nutsvolts.com.**

Repurposing X10 Cameras

Q I have a box of new wireless color X10 security cameras that I purchased in the early 2000s that were a big disappointment as they did not have enough power to transmit to the receiver through the house walls. Is it possible to convert the camera to IoT, Zigbee, or another Wi-Fi standard so that I could actually use these as intended? I also have their pan/tilt bases.

— William Brower
Cadillac, MI

A It seems like X10 cameras of that era are NTSC video cameras. NTSC — National Television Systems Committee — is the analog video standard that was in use in the US and other countries from about the beginning of broadcast television (1941) until it was retired from use in high power broadcast over the air in the US in 2009. As a side note, because of the way that colors were encoded in NTSC using phase encoded subtractions from the luminance signals, there was a lot of variability possible in the decoded colors. Careful adjustment was required to recover the intended color gamut. Thus, engineers that dealt with it affectionately called it Never (Twice) The Same Color. Despite that, the way that color signals were added in a backward compatible way to the original black and white standard was quite ingenious.

There is probably a low power 2.4 GHz transmitter in the camera enclosure that transmits a standard NTSC signal for reception and display on a standard analog television. This transmitter — given that it's an FCC (Federal Communications Commission) Part 15 device — is probably using a very inefficient antenna. One potential way to get more use out of the camera would be to find the antenna feed point and make a better antenna. The downsides are twofold.

First, you can only enhance that antenna so much before the effective radiated power, or ERP, is high enough

• Repurposing X10 Cameras

• The Real Reason They're Called Op-Amps

that the device begins to violate Part 15 rules. If you have an Amateur Radio license and the device falls inside the 2.4 GHz band — 2,300-2,450 MHz — much higher ERP would be permitted as long as the callsign of the licensee was visible in the picture. Second, you would need an analog receiver to view the picture, or a combination receiver and digitizer to make a video bitstream for digital transmission. If there is an included receiver, the receiver antenna could be enhanced without worrying about Part 15. The FCC only cares about the transmitted power.

There are a number of NTSC analog to USB digitizers that are commercially available. For example, Amazon has several in the \$20 range. I just searched for “NTSC digitizer.” USB is simply a hardware interface standard, though, so some kind of software is needed to capture and encode the video from the USB port. That software is typically included with the hardware. Mostly it only works on Windows.

With that, we have enough to make something work. You could either remove the transmitter, routing the NTSC analog signal directly from the camera into the digitizer, but that doesn't allow it to work wirelessly. Another approach would be to use the 2.4 GHz receiver (assuming one came with the cameras) to receive the analog video, and then convert it with one of these digitizers. A PC or Mac could then be used to send the video bitstream out over the Internet, for example.

So, I think at this point you might begin to wonder if this is practical or worth the trouble. If you really want to learn about NTSC video and antennas, I think it would be a fun project. If, however, you are thinking about using these in any realistic application, I think buying a modern IP (Internet Protocol) based camera would be the easy route. X10 still makes them, along with many other manufacturers.

For reference, here is a how-to from someone who converted one of these cameras into a helmet-cam for motorcycling: <http://mods-n-hacks.wonderhowto.com/how-to/turn-wireless-x10-camera-into-helmet-cam-201647>

Post comments on this article and find any associated files and/or downloads at
www.nutsvolts.com/magazine/article/November2016_QA.

The Real Reason They're Called Op-Amps

Q I've always been fascinated by the ability to create both differentiation and integration using just an op-amp. Could you show me how to make both of the circuits using current op-amp technology? I think it would be useful to build both of these types of circuits, then connect them to a microcontroller and view the input/output on my oscilloscope.

Jim H. Parker
Houston, TX

A Operational amplifiers, or op-amps, were originally designed to do exactly what you are asking about: perform mathematical operations. It was an early form of computing using analog circuits. They are easily capable of performing the four basic operations (addition, subtraction, multiplication, and division), but there are other more complicated functions that they can model if we add other components. This includes some elementary calculus as mentioned in your question. It will require a little math, so please bear with me.

It's an interesting question as to why this is possible. The op-amp is just a conduit, if you will, to exploit a characteristic of electric and magnetic fields to do this integrating and differentiating. If you take a look at Maxwell's Equations, and in particular Ampere's Law in integral form (Figure 1), you can see that for a given

$$\oint_C \mathbf{H} \cdot d\mathbf{l} = I + \frac{d}{dt} \int_S \mathbf{D} \cdot d\mathbf{A}$$

■ **FIGURE 1.** Ampere's Law from Maxwell's Equations.

magnetic field (the lefthand side of the equation), current is proportional to the time derivative (time rate of change) of the electric field, summed through some surface, sometimes called the field displacement (the rightmost term). This is what gives us the relationship between voltage and current in a capacitor; the current

being the capacitance times the time derivative of the voltage (Figure 2). This is the derivative that we can use with an op-amp to differentiate a signal.

$$I = C \frac{dV}{dt}$$

■ **FIGURE 2.** Voltage and current relationship for a capacitor.

A capacitor's plates have an electric field between them when a voltage difference is present. When that field changes with time by varying the voltage at the terminals, it becomes the electric field displacement in Ampere's Law. That so-called field displacement is just like a current, and is realized as a current at the capacitor's terminals. The capacitance value of a capacitor is a way of saying how much displacement current will flow for a given time rate of change in voltage. It's effectively saying how strong this effect will be.

A simple circuit to show that relationship is given in Figure 3. I find that an easy way to think about op-amps



EARN MORE MONEY

Get your dream job!

Be an FCC Licensed Wireless Technician!

Make up to \$100,000 a year and more with NO college degree

Get your "FCC Commercial License" with our proven Home-Study Course!

- No costly school. No classes to attend.
- Learn at home. Low cost!
- No experience needed!
- MONEY BACK GUARANTEE:** You get your FCC License or money refunded!



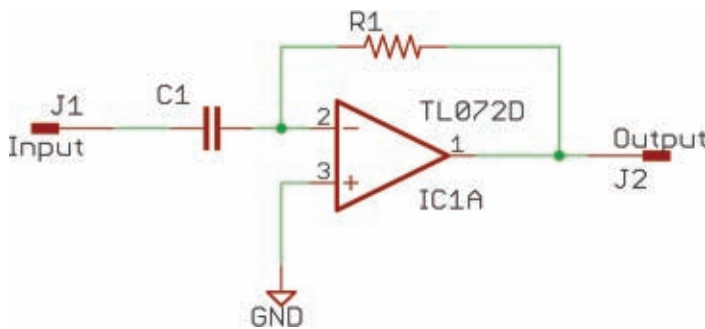
Your "ticket" to 1,000's of high paying jobs in Radio-TV, Communications, Avionics, Radar, Maritime & more.

Call for FREE info: 800-877-8433 ext. 109

www.LicenseTraining.com

COMMAND PRODUCTIONS • FCC License Training
Industrial Center, 480 Gate Five Rd., PO Box 3000, Sausalito, CA 94966-3000

Our
39th
Year



■ FIGURE 3. Active differentiator circuit.

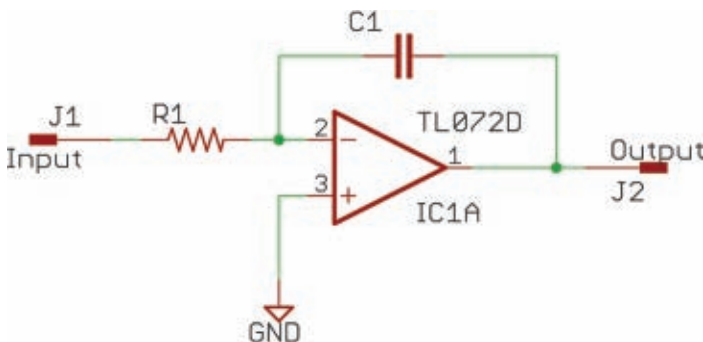
is that when there is negative feedback, they work very hard to keep the + and - inputs at the same voltage. In the case of this circuit, the + input is at circuit ground potential (let's say zero volts relative to the power supplies), so the op-amp will work to keep the - input at that same potential. As we increase the input voltage, the current flowing through the capacitor will be the time derivative (time rate of change) of that voltage.

That would normally cause the - input to rise, but the op-amp wants it to go back down so that it stays the same as the + input. In order to do that, it must match the current flowing through the capacitor with the current flowing through the feedback resistor by decreasing the output voltage in proportion to the capacitor's current.

This follows Kirchhoff's current law if we assume that the op-amp's input current is negligible. So, the op-amp's output voltage will be proportional to the capacitor's current, and so it will be the negative of the derivative of the input voltage. The constants of proportionality will be the capacitor's capacitance and the resistor's resistance. It will follow the relationship in Figure 4, written to equate the two currents.

$$\frac{V_{out}}{R} = -C \frac{dV_{in}}{dt}$$

■ FIGURE 4. Relationship for differentiator circuit.



■ FIGURE 5. Active integrator circuit.

To do integration instead of differentiation, we reverse the roles of the resistor and capacitor as you can see in Figure 5. In that case, the voltage at the output of the op-amp must make the current through the capacitor match the current flowing through the input resistor. The only way to do that is to essentially invert the voltage-current relationship of the capacitor. This gives us an integral instead of a differential.

The integration is considered over some time interval (see Figure 6 for the relationship). Note that there is a dual (like a mirror image) of this circuit that can be made with an inductor. Inductors have a similar time derivative relationship to a capacitor, with the roles of voltage and current reversed.

$$\frac{V_{out}(t_1) - V_{out}(t_0)}{R} = -C \int_{t_0}^{t_1} V_{in} dt$$

■ FIGURE 6. Relationship for integrator circuit.

To observe this on an oscilloscope, all you need to do is provide an input waveform. Many of the available microcontrollers (like an Arduino) have a PWM, or pulse width modulated output that can be used with a filter to produce sinusoidal AC waveforms.

That's not ideal for this application, though, since you can't observe other shapes and how they're changed by the circuits.

Some Arduinos also have a DAC (digital-to-analog converter) output, so that arbitrary waveforms can be reproduced with a table of samples. There is an Arduino tutorial that describes a waveform generator at <https://www.arduino.cc/en/Tutorial/DueSimpleWaveformGenerator>.

It uses a Due, which is unfortunately no longer available. There are several newer boards that will work, though. Look at the table on their website that can be found at <https://www.arduino.cc/en/Products/Compare>.

There is a column labeled "Analog In/Out." Choose one that has at least one output. Then, you can choose values for R and C that will give you an observable output using the equations provided. As for op-amps, the figures use a TL072, but any common device you might have in the parts drawer should work. Just give it a ±15V supply if that's what the device wants. **NV**

Give the gift of Nuts & Volts to a friend and get a FREE gift for yourself. See page 5 for details.

2-For-1 Holiday Offer

Give a one year (12 issues) subscription to a friend and receive a **FREE** one year subscription for another friend, or yourself!



Not only will they get a full year of SERVO delivered right to their door, they'll also get free access to our digital (web) edition, PLUS our mobile (iOS, Android, Kindle Fire) edition.

Subscribe online at:
www.servomagazine.com



Be sure to use promotion code
Y6WXMSA when ordering



NEW PRODUCTS

- HARDWARE
- SOFTWARE
- GADGETS
- TOOLS

AC/DC FOUR PORT MULTICHARGER

Building on the popularity of their X4 AC Plus, Hitec RCD USA has taken things up a notch with the introduction of their X4 AC Pro high performance microprocessor controlled charger/discharger with extensive battery management capabilities. Capable of operating on 100-240 volts AC household current or from an 11-8 volts DC power source, this new multifaceted powerhouse exudes a total charge circuit power of 200 watts with power distribution capability in AC mode and a 300 watt output when in DC mode.

The X4 AC Pro allows users to connect up to four batteries of varying chemistries (NiMH / NiCd / LiPo / LiFe / Lilon / LiHV / Pb) to any of the charging channels with each channel performing independently of one another.

Additional features include Hitec's "Charge Master" software, providing the ability to operate a charger via a PC, plus a built-in Bluetooth for smartphone app operation. The estimated street price for the X4 AC Pro AC/DC multicharger is \$199.99.

Charging specifications include:

- AC input: 100-240 volts AC
- DC input: 11-18 volts DC
- Total charge circuit power on AC power: 200 watts with power distribution
- AC power distribution
 - Channels A & C: 100 watts total
 - Channels B & C: 100 watts total
- Power distribution adjustable range
 - Channels A & B: 50-100 watts
 - Channels C & D: 0-50 watts
- Total charge circuit power on DC power:
 - 300 watts total power
 - Channels A & B: 100 watts each
 - Channels C & D: 50 watts each
- Charge Current



- Channels A & B: 0.1-10.0 amps
- Channels C & D: 0.1-5.0 amps
- Discharge wattage: 10 watts per channel
- Discharge current: 0.1-2.0 amps per channel
- Discharge current for Lithium battery balance: 200 mAh
- LiPo, LiHV, LiFe, and Lilon cell count: 1-6 cells
- NiMH / NiCd cell count: 1-15 cells
- Pb / Lead acid voltage range: 2-20 VDC
- Battery capacity range: 100-50,000 mAh
- Auxillary output: 5V 2.1A DC USB device charge port
- Communication methods
 - USB port for PC control and firmware upgrades
 - Bluetooth 4.0 for smartphone apps
- Dimensions: 3.8 x 7.2 x 2.8 in / 97 x 182 x 71 mm
- Weight: 47 oz / 1,335 g

For more information, contact:
Hitec RCD USA, Inc.
www.hitecrd.com

WIRELESS REMOTE CONTROL OUTLET

TUDIA's Cous wireless remote control electrical outlet is an

inexpensive way to control lights, household appliances, and power strips without the need for expensive IoT (Internet of Things) smartphone devices. TUDIA Cous turns on and off lights and appliances, wirelessly communicating to its outlet plugs

from as far away as 100 feet — even through closed doors, floors, and walls. Made for use with all hard-to-reach lights and appliances, and essential gear for the mobility impaired, Cous comes with five outlet plugs and two remotes



(additional outlet plugs and remotes are available). The Cous outlet plug features a compact size that covers only one wall outlet when plugged into the top, freeing up the lower outlet for other uses (unlike others that cover two outlets at once).

TUDIA Cous is extremely easy to use. Simply plug in one of the five small outlet plugs into a regular wall outlet to turn appliances and lights on and off with the included two pre-programmed remote controls. Each remote can control up to five devices, whether an appliance is missing a built-in on/off switch or is simply in a hard-to-reach place.

Cous also adds an eco-friendly advantage by cutting back on energy vampire drain caused by keeping cable boxes, blenders, toasters, microwaves, TVs, etc., plugged in, saving up to 10% monthly on energy bills while also helping the environment.

Leaving lights and appliances turned on all the time also makes electronics wear out faster, so with Cous, the life of appliances can be extended up to 15%.

In a nutshell, TUDIA Cous features include:

- Easy installation: Simply plug in to turn appliances on and off



- with remote control.
 - Remote signal communicates to outlet plug through closed doors, floors, and walls from as far away as 100 feet.
 - Provides assistance for the mobility impaired.
 - Turns on/off hard-to-reach lights and appliances.
 - Single remote can control multiple devices or multiple remotes can control a single device.
 - Turning off appliances when not in use saves up to 10% in monthly energy bills and helps devices last up to 15% longer.
 - Low cost: Priced at \$34.74 for five plugs and two remotes.
- Specs are as follows:
- Power input: 120V/60 Hz
 - Power output: 1,200W/10A (max)
 - Transmission frequency: 433.92 MHz
 - Remote transmitter battery: DC 12V(23A) 1pc
 - Power consumption: 0.6W

The Cous package includes two 12 volt batteries and a user's manual.

For more information, contact:
TUDIA, Inc.
www.tudiaproducts.com

Continued on page 60



The arLCD 3.5" smart TFT touchscreen combines Arduino UNO with the ezLCD smart LCD.

earthlcd.com/nuts

ROBOT POWER
Extreme Motor Speed Control!

MegaMoto GT- Mega Motor Control for Arduino™

- 6V-40V - 35A / 50A+ peak
- Single H-bridge or dual half
- Current and Temp protected
- Jumper select Enable & PWM
- Robust power terminals

The VYPER™

- 8V-42V - 125A / 200A peak!!
- Single H-bridge
- RC and Pot control
- Overload protected
- For your BIG bots

www.robotpower.com
 Phone: 360-515-0691 • sales@robotpower.com

SDP Stock Drive Products
 Setting Ideas Into Motion

One-Stop Shop for
 Mechatronic Components

EXPLORE
 DESIGN
 BUY ONLINE



www.sdp-si.com
 no minimum requirement

Designatronics Inc.

ELECTRONICS FOR KIDS

New in the Nuts & Volts webstore!

If you've ever looked at an electronic gadget and thought, "How does that work?" or "How can I make that?" Then look no more.

Whether you're 8 or 100 years old, as long as you're curious and have a playful spirit, this 328 page book is for you.

For complete details and to view a sample chapter from the book, visit our webstore @ store.nutsvolts.com

Only \$24.95

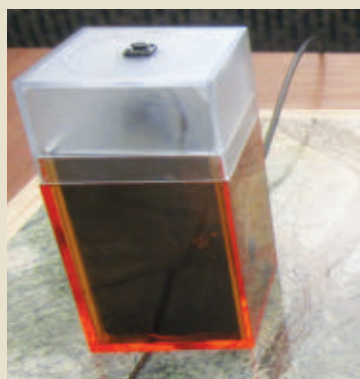
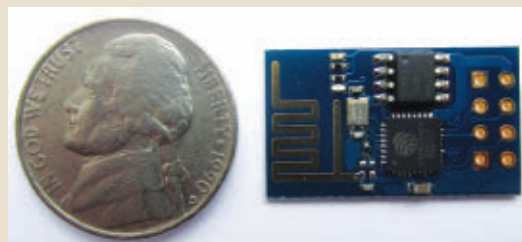
ESP8266 WEATHER CLOCK

By Craig A. Lindley

Post comments on this article
and find any associated files
and/or downloads at
[www.nutsvolts.com](http://www.nutsvolts.com/magazine/article/November-2016_ESP8266-Weather-Clock)
/magazine/article/November-
2016_ESP8266-Weather-Clock.

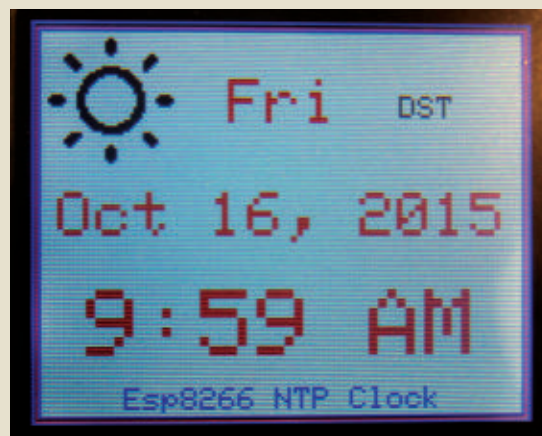
Readers of *Nuts & Volts* may recall my three previous articles about using the amazing ESP8266 family of devices. They are:

1. "Meet the ESP8266: A Tiny, Wi-Fi Enabled, Arduino Compatible Microcontroller" in the October 2015 issue.



2. "Thinking of You" article in the November 2015 issue.

3. "ESP8266 NTP Clock" article in the June 2016 issue.



The more I use these devices, the more I am impressed with their value proposition, capabilities, and robustness. After writing the ESP8266 NTP Clock article where I coupled a NodeMCU Amica module (which contains an ESP8266-12 device) with an Adafruit 1.8" TFT LCD display, I started to look around for other applications that I could use this same hardware for. I came up with two ideas:

1. Building a mini weather station that uses weather data available on the Internet (in this case, from **myweather2.com**) for any location in the world and displays it on the LCD display.
2. An RSS Feed reader that can display headlines from various news sources across the Internet on the LCD display.

In this article, I will present the implementation of the first idea in what I call my Weather Clock which combines the display of localized weather conditions with the auto setting NTP clock from my previous article. So, with extremely simple hardware with a low parts count, you can get current and forecasted weather conditions for your specific location, plus have a clock that never needs to be manually set. Pretty sweet, don't you think?

I'll save the RSS Feed reader (which is really cool as well) for a future article. This also runs on exactly the same hardware as the weather clock and the NTP clock.

Building a weather station with a microcontroller is hardly breaking news, but most of the implementations I have seen used a PC or Raspberry Pi to access the

Internet for the weather data, then messaged the data into a small enough package that it could be transferred to the microcontroller weather station for display. In other words, the microcontroller based system was just a display for the previously digested weather data.

I took this as somewhat of a challenge to see if I could combine both weather data acquisition and display using a single ESP8266 device and — while I was at it — see if I could include the NTP clock functionality as well.

I'm pleased to say that I was able to pull this off. If you have ever wanted to build a mini weather station/clock for your home or business, I don't believe you will find a simpler or cheaper solution than the one presented here.

Hardware

As mentioned, the weather clock uses the same hardware as incorporated in my ESP8266 NTP clock. To save you from going back and (re)reading the previous article, the hardware information is repeated here, starting with the minimalist **Parts List**.

Figure 1 shows a Fritzing connection diagram/schematic for the weather clock. **Figure 2** shows the design wired up and working on a breadboard. NOTE: There isn't a wire color correlation between **Figures 1** and **2**. As shown, the weather clock is powered via a USB cable and a USB power supply module.

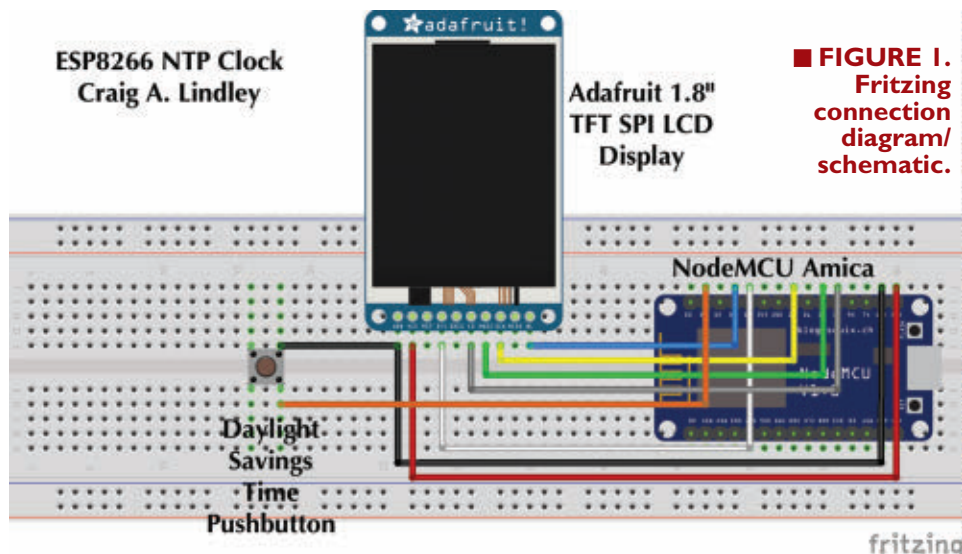
The wire by wire connections are shown in **Table 1** because they might not be clear from the Fritzing diagram.

The GPIO designations are shown in **Table 1** as that is how these digital I/O lines are referred to in the Arduino code. The Adafruit LCD display has a micro-SD memory card connector and interface which can be used with the ESP8266, although they were not needed for this project.

Software

The software for the ESP8266 weather clock was developed using the Arduino IDE (integrated development environment). Refer to my previous articles and/or the Resources section for how to set up the Arduino IDE on your computer for targeting ESP8266 type devices. Make sure to select "NodeMCU 1.0 (ESP-12E Module)" as the board type in the tools menu.

The ESP8266 weather clock software is available at



■ **FIGURE 1.**
Fritzing
connection
diagram/
schematic.

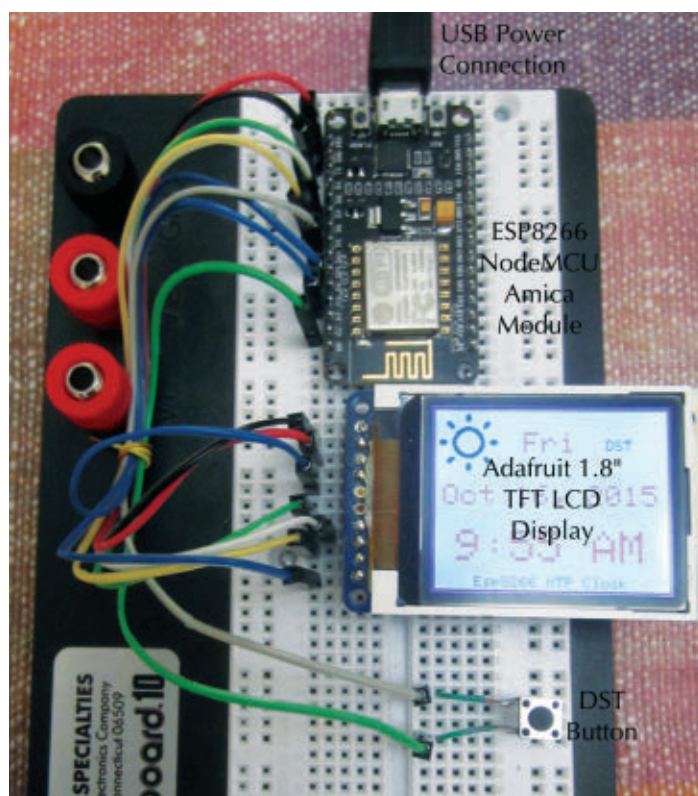
PARTS

NodeMCU LUA Amica R2 Module
1.8" TFT SPI LCD Display (black tab)
Pushbutton Switch SPST
USB Cable — USB A to USB Micro B
USB Power Supply
(capable of at least one amp at five volts)
Hook-up wire and breadboard

SOURCE

Electrodragon.com
Adafruit.com; Product ID 358
RadioShack or anywhere else
RadioShack or anywhere else
RadioShack or anywhere else
RadioShack or anywhere else

PARTS LIST



■ **FIGURE 2.** The design wired up and working on a breadboard.

Table 1.		
NodeMCU Amica Pin	Adafruit 1.8" Display Connection	DST Pushbutton SPST Switch
D1 (GPIO 5)		SW1
D3 (GPIO 0)	LITE	
D4 (GPIO 2)	D/C	
D5	SCK	
D7	MOSI	
D8 (GPIO 15)	TFT_CS	
3V3	VCC	
GND	Gnd	SW2

the article link. (The file is called `Lindley_ESP8266WeatherClock.zip`.) To use this software, unzip it and copy/move the `ESP8266WeatherClock` directory from the zip file into your Arduino directory. While the hardware is about as simple as possible, the software is quite complex, and is made up of the files shown in **Table 2**.

In addition to the files in **Table 2**, **Table 3** shows the Arduino libraries which are also required.

The version of these libraries I used to develop the weather clock are also included in the zip file for this article. Remember libraries must be installed in the `arduino/libraries` directory on your development computer, and the Arduino IDE must be restarted to recognize them.

User Configuration of the Weather Clock Software

The weather clock's software must be configured before it will work correctly. All user configuration items are found in the `ESP8266WeatherClock.ino` file. Please locate the following text in that file:

```
// *****
// Start of user configuration items
// *****

// Set your WiFi login credentials
```

Information about myweather2.com's weather data feed
www.myweather2.com/developer.

Information about JSON
www.json.org

Information about NTP
www.ntp.org

Information about programming the ESP8266 in the Arduino environment

Resources

github.com/esp8266/Arduino
(plus in my three articles mentioned previously)

Information about the NodeMCU Amica
www.electrodragon.com/product/nodemcu-lua-amica-r2-esp8266-wifi-board

Information about the Adafruit 1.8" TFT SPI LCD display
www.adafruit.com/products/358

```
const char * WIFI_SSID = "xxxxxxxxxx";
const char * WIFI_PASS = "xxxxxxxxxxxx";

const int     TIMEZONE_OFFSET = -7;
              // Set your timezone offset (-7 is
              // mountain time)
const bool    USE_DST = true;
              // Set to false to disable DST mode
const bool    HOUR_FORMAT_12 = true;
              // Set to false for 24 hour time mode

const char * API_KEY = "yyyyyyyy";
              // Key/Access code from myweather2.com
const char * LOCATION_STRING = "zzzzzz";
              // Location indicator

// *****
// End of user configuration items
// *****
```

First and most importantly, you must modify the code with the SSID and password of your Wi-Fi network. Otherwise, the weather clock won't be able to access the Internet (and by extension, the weather data feed from **myweather2.com**) nor the NTP servers that provide the time.

Next, you must set the timezone offset for your location to make the clock display the correct time.

Timezone offsets can be found at https://en.wikipedia.org/wiki/List_of_UTC_time_offsets.

You must next indicate whether your clock will use daylight saving time or not, and whether it will operate in a 12 or 24 hour format. `USE_DST` must be set true if your clock will use daylight saving time whether or not DST is currently in effect. Set `HOUR_FORMAT_12` true to run your clock in the 12 hour format; otherwise, it will operate in the 24 hour time format.

You must get an API key from **myweather2.com** (which they refer to as the Unique Access Code) to access their service and retrieve weather data from them. To get a key/code, go to www.myweather2.com/ilogin.aspx and fill out their registration form. After you have created an account, you need to log in and go to your Developer Zone page; you will see the key/code they have assigned you. This key/code must be transferred to the `API_KEY` entry in the configuration data shown previously. Note, only one key/code is available per email address.

You must also tell the **myweather2.com** service the location you want the weather data for. This is what the `LOCATION_STRING` in the configuration data does. This string has three possible formats:

1. A UK post code
2. A US zip code
3. Latitude/longitude

For my weather clock, I used my zip code.

The code can be compiled and uploaded to the NodeMCU Amica device once the configuration data is set and all of the required libraries have been installed in the Arduino environment.

Weather Clock Operation

Figures 3 through 9 show the weather clock in operation. Each of these images typify what I refer to as a display page. Each display page shows different information (refer to Table 4), but all pages have a series of seven small circles at the bottom of the page to indicate which display page is currently being shown.

When the weather clock software first starts, there is (usually) a short delay while three things happen. First, the weather clock logs into the local Wi-Fi network. Next, the NTP clock code initializes and makes a request over the Internet to retrieve the time from an NTP time server. Finally, weather data is requested and retrieved from **myweather2.com**. If all is well, the first display page with the Wi-Fi icon and credits is selected (Figure 3).

After a programmable length of time, the other six display pages are sequentially shown. After display page 7 (the NTP clock) is displayed, the sequence repeats starting with display page 1.

By default, time data is retrieved every five minutes; weather data is retrieved every 15 minutes; and display pages are changed every 12 seconds. Of course, each of these time intervals can be changed in the software. I should point out that the acquisition of weather and time data is completely disjoint from its display. That is, these two processes happen completely independent of each other.

The operation of the weather clock can best be understood by examining the code in the file *ESP8266WeatherClock.ino*. A Finite State Machine (FSM) contained in the *loop()* function orchestrates everything. Technically, an FSM is:

“A model of a computational system, consisting of a set of states, a set of possible inputs, and a rule to map each state to another state, or to itself, for any of the possible inputs.”

Sounds daunting, but don't let the definition scare you. The operation is really quite simple. The weather clock's FSM is defined by this series of states:

INIT, CHECK_EVENTS, ACQUIRE_DATA, ADVANCE_DISPLAY, UPDATE_DST_STATUS

Table 2.

File	Description
DisplayPages.h	Code for each of the seven display pages.
ESP8266_ST7735.cpp	LCD driver code specific to the Adafruit 1.8" (black tab) display utilizing the hardware SPI interface of the ESP8266.
ESP8266_ST7735.h	Header file for the LCD driver code above.
ESP8266WeatherClock.ino	Main program. Initializes the hardware, logs into the local Wi-Fi network, and then installs the NTP code as the time provider. It then manages the display of the pages on the LCD.
Icons.h	Data for the Wi-Fi, sun, and moon icons. Data is in xbm format.
Misc.h	Code for reading and writing the ESP8266's EEPROM.
NTP.h	Functions for sending UDP packets to NTP servers and retrieving the GMT time and converting it to local time.
TGFunctions.h	Misc functions for formatting text and graphical data for display on the LCD.
Weather.cpp	Weather class for sending weather data requests to myweather2.com for retrieving the JSON data stream returned, and then parsing the data to extract the pertinent weather attributes for display by the various display pages.
Weather.h	Header file for the weather class above.

Table 3.

Library	Function	Source
Adafruit_GFX NOTE: This library had to be modified for use with the ESP8266, so it is not the stock library.	Text and graphics functions for the LCD display driver	https://github.com/adafruit/Adafruit-GFX-Library
Time	Updated and improved version of the Arduino library	https://github.com/PaulStoffregen/Time
ArduinoJson	JSON parser	https://github.com/bblanchon/ArduinoJson

Table 4.

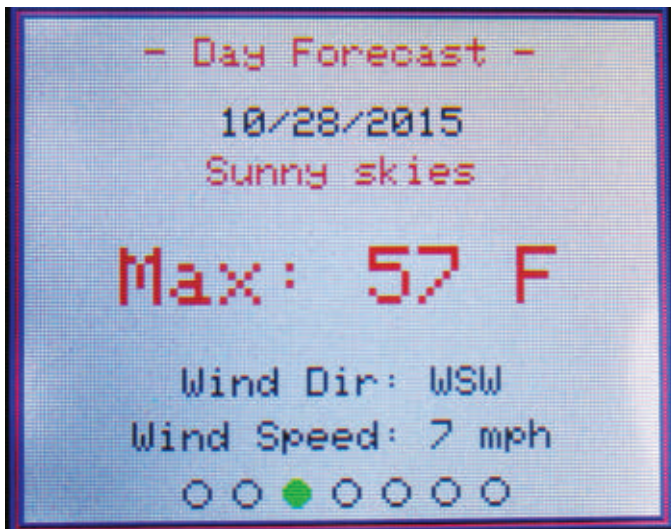
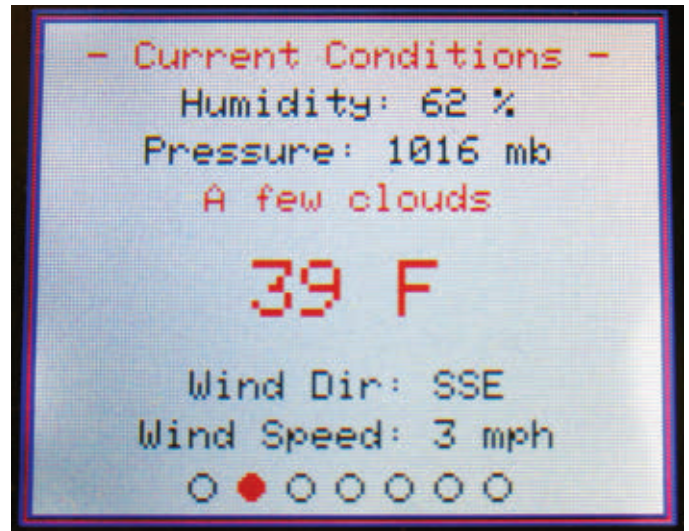
Display Page Number	Information Displayed
1	Wi-Fi Logo and Credits
2	Current Weather Conditions
3	Day Conditions Forecast
4	Night Conditions Forecast
5	Next Day Conditions Forecast
6	Next Night Conditions Forecast
7	NTP Time and Data Display

which will each be described shortly. In addition, there is a variable called *state* which tells the state machine which state it is currently in. Transitions between the states happen when certain events or inputs occur. The FSM is guaranteed to be in one of the defined states every time through the *loop()* function.

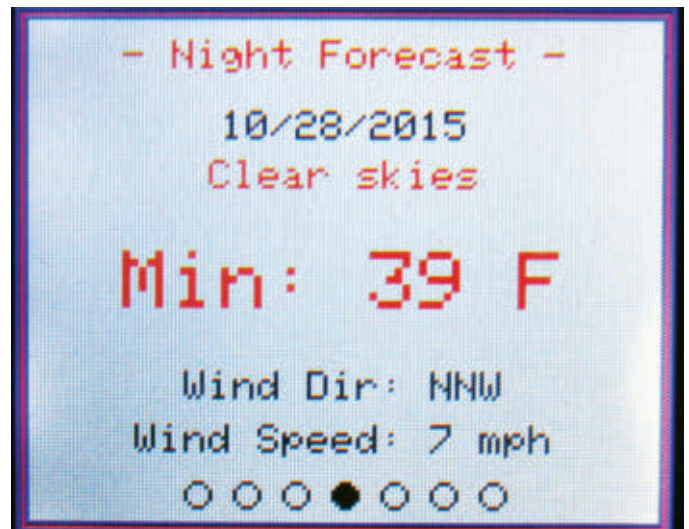
■ **FIGURE 3. Display page 1: Wi-Fi login and program credits.**



■ **FIGURE 4. Display page 2: Current weather conditions.**



■ **FIGURE 5. Display page 3: Day conditions forecast.**



■ **FIGURE 6. Display page 4: Night conditions forecast.**

The *INIT* state is the initialization state for the state machine and it is only entered once when the ESP8266 device is first powered up. In this state, the display page number variable is set to one; display page 1 is shown on the LCD, and the *dataAcquisitionCount* and the *displayAdvanceCount* variables are initialized to a time in the future when new weather data is to be acquired and when the next display page is to be shown. The *CHECK_EVENTS* state is then selected for the next iteration of the *loop()* function.

The FSM will stay in the *CHECK_EVENTS* state until one of the following things happen:

1. Activity is detected on the DST pushbutton switch.
2. The weather data acquisition count has expired and new weather data needs to be acquired.
3. It is time to advance to the next display page.

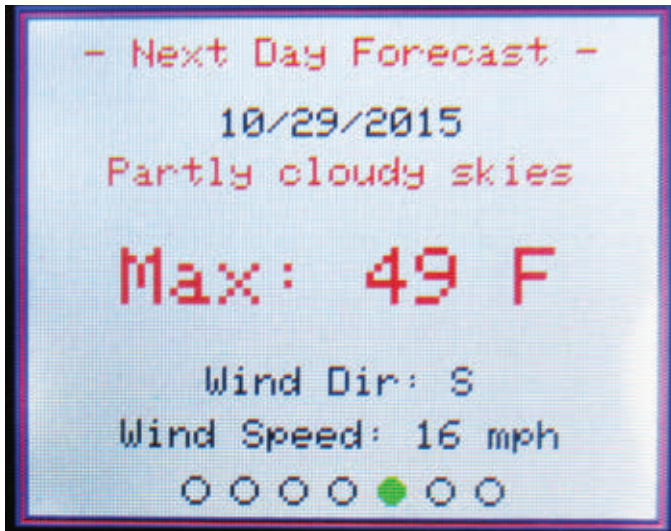
The state of the FSM will change to *UPDATE_DST_STATUS* if number one occurs; it will change to *ACQUIRE_DATA* if number two occurs; and it will change to *ADVANCE_DISPLAY* if number three occurs.

In the *UPDATE_DST_STATUS* state, the daylight saving time (DST) status is toggled. If DST was on, it is turned off; if it was off, it is turned on. Every change to the DST state is stored in the ESP8266's EEPROM so that it survives power outages.

The display page variable is then set so that display page 7 (the NTP Time and Date page) will be displayed the next time the *ADVANCE_DISPLAY* state is entered. Finally, the state is changed to *ADVANCE_DISPLAY* for the next trip through the loop.

In the *ACQUIRE_DATA* state, the *retrieveWeatherData* function in the *Weather* class is called to acquire new weather data. A call to this function causes a whole chain

■ FIGURE 7. Display page 5: Next day forecast.



of events to occur. First, an HTTP GET request is built up using the *API_KEY* and the *LOCATION_STRING*, and it is sent to **myweather2.com**.

The returned weather data in JSON format is stored line by line in a buffer for later processing. JSON or (Javascript Object Notation) is a text-based human-readable data interchange format used for representing simple data structures and objects in web browser based code. See **Resources** for information about JSON if you are interested.

After all the data has been retrieved, it is passed to the *ArduinoJson* parser which makes the various data items easily accessible. All of the weather attributes displayed on the five weather pages are extracted from the returned JSON data.

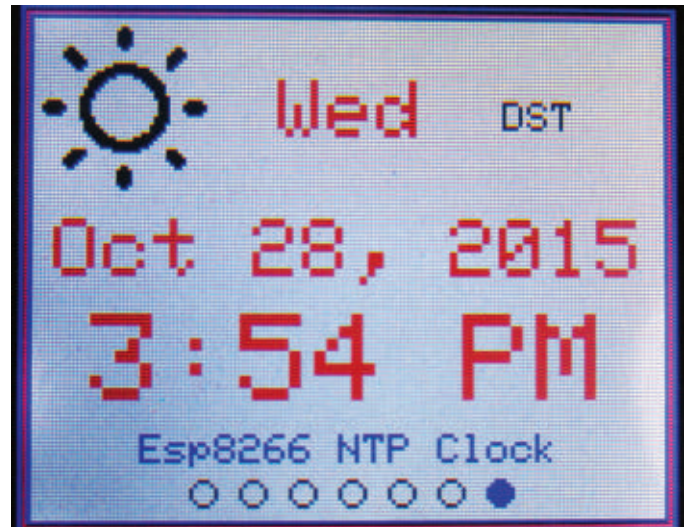
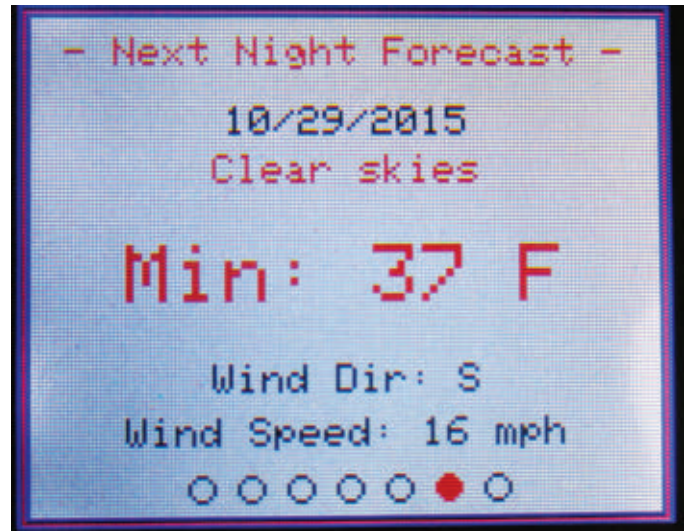
Once new weather data has been acquired, the *dataAcquisitionCount* variable is reinitialized to a time in the future when new weather data will again be required. Finally, the state is changed to *CHECK_EVENTS* for the next pass through the loop.

The *ADVANCE_DISPLAY* state is entered when it is time to change the displayed page. In this state, the LCD display is cleared and the frame which is common to all display pages is drawn around the perimeter of the display. The *displayPageNumber* variable is then incremented and wrapped around if necessary and the new display page is shown.

Finally, *displayAdvanceCount* is reinitialized and the state variable is set back to *CHECK_EVENTS* for the next pass through the loop. In case it is not obvious, the ESP8266 spends most of its time spinning in the *CHECK_EVENTS* state — only changing states occasionally when one of the three events occur.

Astute readers may be wondering how and when the NTP time data gets updated since there are no references to *time update* in the FSM. That is because *time update* is

■ FIGURE 8. Display page 6: Next night forecast.



■ FIGURE 9. Display page 7: NTP time and date display.

handled behind the scenes in the Time library and so doesn't need to be explicitly performed in the weather clock's FSM code.

Last Thoughts

To build a weather clock of your own, connect the Adafruit 1.8" LCD display to the NodeMCU Amica module and then connect a DST pushbutton switch. Next, connect a USB cable from the NodeMCU Amica module to your development computer. After installing the required libraries, bring up the Arduino IDE and load the software for this article. Edit the User Configuration data as described previously, and then compile and upload the code.

If you did everything correctly, you should have a fully functional stand-alone weather clock of your own. **NV**

A 3-in-1 Tester for YOUR LAB

By Jim Teixeira

Post comments on this article and find any associated files and/or downloads at www.nutsvolts.com/magazine/article/November2016_3-in-1-Lab-Test-Equipment

This article describes a piece of lab test equipment that combines a variable DC voltage reference, a programmable function wave generator, and a fully functional frequency counter. This project started as a microcontroller

controlled variable voltage reference (VVR) using just a PIC, a fixed voltage reference, and a digital-to-analog converter (DAC). The project then expanded to include a programmable function generator (PFG) that can produce a sine wave, triangle wave, square wave, etc. I also needed a mechanism to measure the frequency of the function generator output, so I added a frequency counter capability to the PIC. The total parts cost of this multifunction tester is less than \$150 (includes components, power supply, cabinet, all front panel parts, printed circuit boards [PCBs], etc.).



The key component of the design is the eight-bit DAC0800 digital-to-analog converter shown in **Figure 1**. The analog output of the DAC0800 is connected to the LM6181 op-amp similarly to that shown in Figure 26 of the Analog Devices Application Note AN-17. The DAC0800 digital input can be connected either to the eight-bit PIC port D or to the input/output (tri-state) buss of an eight-bit RAM. As I describe the design of the tester, the interaction between the components will become clearer.

At the top left of **Figure 1** is a 10 volt (I actually use two five volt diodes in series) voltage reference that determines the DAC maximum output voltage. When in the function generator mode, a front panel level adjust potentiometer sets the actual peak-to-peak output voltage. When in the VVR mode, the potentiometer is bypassed and the PIC sets the output level. The LM6181 is a high performance amplifier that can output ± 10 volts at ± 100 milliamps.

In the VVR mode, the LM6181 input connection is set for unipolar output by grounding the plus input of the

LM6181 (see AN-17); the PIC port D is set as an output; and the RAM tri-state output is turned off so the PIC has complete control of the DAC0800 output.

If port D is set to binary 00000000, the LM6181 output will be zero volts; if port D is set to 11111111, the LM6181 output will be +10 volts. Since there are 256 possible binary numbers between 00000000 and 11111111, there are also 256 possible output voltages.

The PFG mode can be implemented this way by having the PIC repeatedly output the values describing the function (for example, the sine wave amplitude values), but the maximum frequency attainable will be limited because the PIC — in addition to outputting function values — must also continuously read the front panel switches and perform the frequency counter work.

In my design, I decided to offload the PFG DAC updates to external circuitry: the eight-bit RAM (U4), the eight-bit counter (U3), and the high speed oscillator. This PIC is no longer in the repetitive PFG loop and it can perform the housekeeping duties mentioned above.

We finally get to the frequency counter. Here, I implement a technique I used in an earlier *Nuts & Volts* article on frequency counters (March 2015). The common hobbyist frequency counter design sets up a fixed gate interval (say, one second) and then counts the number of input cycles during that interval. With a one second gate, the number of input cycles is equal to the input frequency with a ± 1 cycle resolution. This works very well if the input frequency is high because the ± 1 count error can be ignored. However, at low frequencies, the ± 1 count error can be very significant.

To accurately measure a low input frequency, the gate interval must be increased, or the counter can measure the period of the input signal. In my design, I automatically switch to a period measurement when the input frequency is less than 150,000 Hz. This is discussed in detail below.

Variable Voltage Reference

In the VVR mode (see **Figure 2**), the PIC sets the VVRMode bit which — with relay K1 — configures the DAC0800/LM6181 pair for zero volts output when the PIC outputs binary 00000000 on port D, and +10 volts for binary 11111111.

If we use 00000000 as a starting point, there are then 255 voltage steps to get to +10 volts. Therefore, each step is $10/255$ or 0.0392 volts. This is an awkward value since it leads to non-integer values. For example, if we want a 5.2 volt output, we need a buss digital value of $5.2/0.0392$ or 132.6.

Rounded to 133 it produces 133×0.0392 or

5.216 volts — a .3% error.

I get rid of this error by simply using 250 as the number of steps to 10 volts, so the voltage per step is $10/250$ or exactly 0.040 volts. Now, the 5.2 volt setting is $5.2/0.040$ or 130, exactly 5.2 volts. This works for all voltages that have an even number for the fractional part (5.0, 5.2, 5.4, etc.).

With an odd fractional part like 5.3 volts, the setting is $5.3/0.0400$ or 132.5, and all voltages with an odd fractional value will have a half bit remainder. I correct this error by using a *HALFSTEP* control line on the PIC which effectively adds a ninth bit to the DAC0800. This is shown at the right in **Figure 2**. If *HALFSTEP* is set high, the 2.4

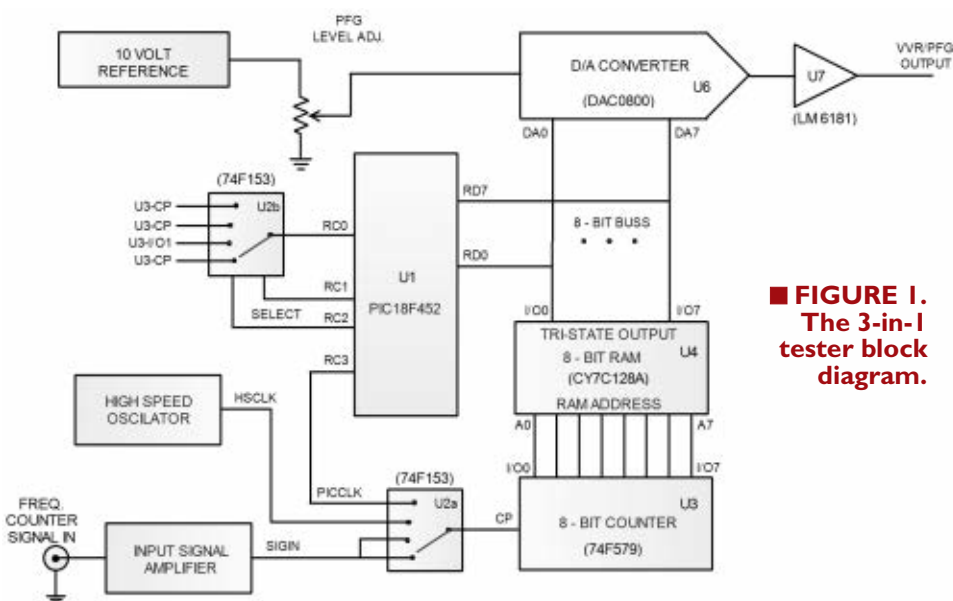


FIGURE 1.
The 3-in-1
tester block
diagram.

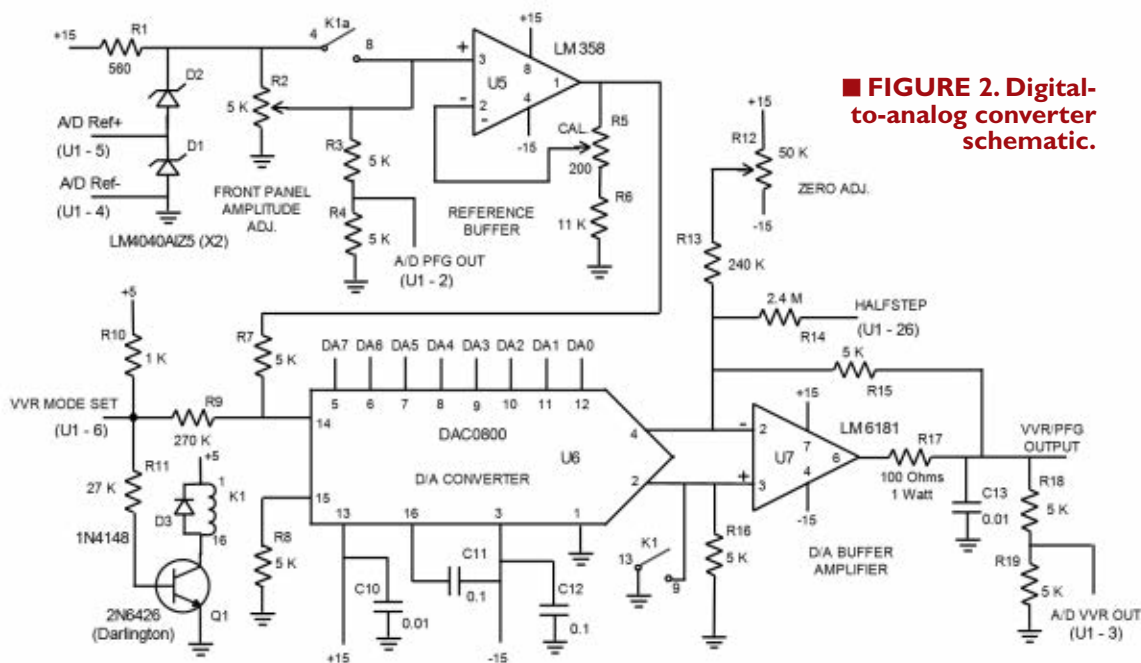


FIGURE 2. Digital-
to-analog converter
schematic.

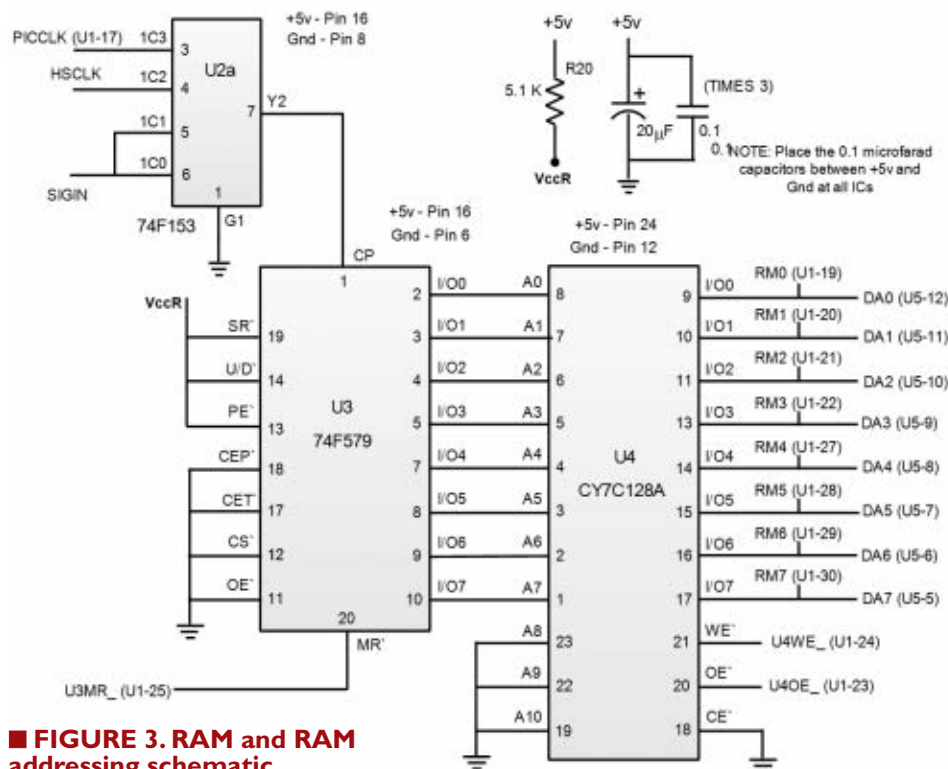
The VVR mode is selected when the PIC sets the *VVRMode* bit. Two things happen: First, the double pole relay K1 is energized and contact K1a bypasses potentiometer R2, connecting the full 10 volts to amplifier

Calibration of the VVR/PFG does not require any external equipment because of the accuracy of the PIC ADC coupled with the accurate five volt reference diode D1. Just follow these steps:

1. Set the *VVRMode* bit and clear the *HALFSTEP* bit.
2. Set SW2 to zero volts and SW3 to .2 volts.
3. Adjust the *ZERO ADJ*, internal potentiometer R12, until the display reads 0.2 volts.
4. Set SW2 to 10 volts and SW3 to zero volts.
5. Adjust the *AMPLITUDE ADJ*, internal potentiometer R5, until the display reads 10 volts.
6. Repeat steps 2 through 6 a few times.

Programmable Function Generator

The PIC18F452 that I chose for this project has a PLL that multiplies the input (10 MHz) by four to 40 MHz, but the instruction rate is one quarter of that or 10 MHz. This means the updates can't take more than four



■ **FIGURE 3. RAM and RAM addressing schematic.**

instruction cycles each. At the same time, the PIC has to monitor all of the front panel switch settings (using the built-in ADC), drive the display, and handle the frequency counter function (discussed below). Using the interrupt control to do the updates just makes matters worse because of the time spent processing interrupts.

As I mentioned in the opening remarks, I decided to offload the DAC updates to external circuitry (see **Figure 3**): the eight-bit RAM (U4), the eight-bit counter (U3), and the high speed oscillator. When you select a function (for example, a sine wave), the PIC loads the RAM with the 256 amplitude constants that define one complete cycle of the sine wave. The PIC then disconnects from the RAM and goes about performing the other tasks. The high speed oscillator is connected to the eight-bit counter via the multiplexer U2a and the RAM continuously cycles

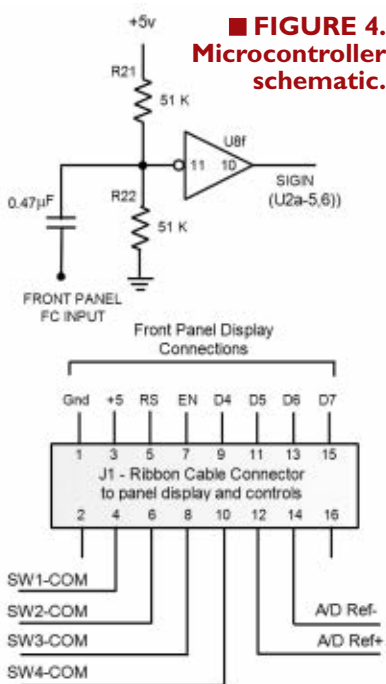
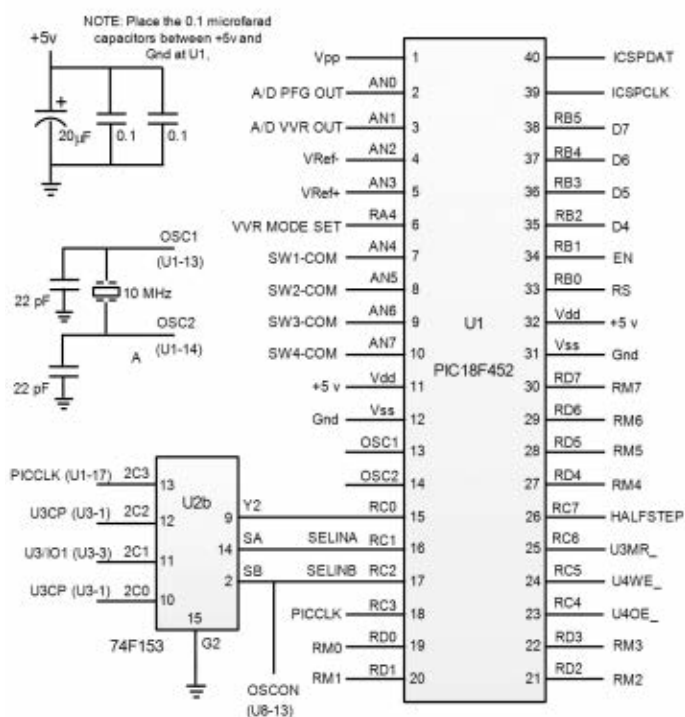
through the DAC steps; out comes a sine wave. The user can program any function: square wave, triangle wave, pulse, etc., to go into the RAM.

Sixty-four steps per quadrant is very high resolution and is probably more than required. ON-OFF-ON toggle switch SW4 sets the resolution at 16, 32, or 64 steps per quadrant. For 32 steps, the RAM is loaded with two complete cycles of the function using every other amplitude step from the 64-bit step list. This also doubles the output frequency. For 16 steps per quadrant, four complete cycles in the RAM are used.

Loading the RAM data is a little bit tricky because the PIC has no direct connection to the RAM address buss (A0 through A7 on U4). So, I have to go through the eight-bit counter U3 to get to the RAM address buss. U2a sets the clock input on U3 to PICCLK — an output pin on

<u>ITEM</u>	<u>DESCRIPTION</u>	<u>SOURCE</u>
R2	5K ohms	Jameco 255514
R3, R4, R7, R8	5K ohms, 1/8 watt, 0.1%	Mouser 71-RN60C5001B
R15, R16, R18, R19		
R5	Bourns 3296W, 200 ohms	Jameco 1156787
R12	Bourns 3296W, 50K ohms	Jameco 853791
R17	100 ohms, 1 watt	Mouser 5073NW100R0J
R22	500 ohms	Jameco 255477
R23	2K ohms	Jameco 255493
U1	PIC18F452	Jameco 248031
U2	74F153	Mouser 595-SN74F153N
U3	74F579	eBay
U4	CY7C128A	Jameco 242691
U5	LM358	Jameco 23966
U6	DAC0800	Jameco 831859
U7	LM6181	Jameco 301874
U8	74AHC04	Mouser SN74AHC04N
C40	47 pF	Mouser FG28C0G1H470JNT6
C41	100 pF	Mouser FG26C0G2J101JNT6
C42	330 pF	Mouser FG28C0G1H331JNT6
C43	.001 μ F	Mouser FG28C0G2A102JNT6
C44	.0022 μ F	Mouser FG18C0G1H222JNT6
C45	.0047 μ F	Mouser FG18C0G1H247JNT6
C46	.01 μ F	Mouser FG18C0G1H103JNT6
C47	.022 μ F	Mouser 5989-400V.022-F
C48	.047 μ F	Mouser 5989-250V.047-F
C49	.1 μ F	Mouser FG16C0G1H104JNT6
D1, D2	LM4040AIZ5	
D3	1N4148 diode	Mouser 926-LM4040AIZ5.0NOPB
SW1, SW3, SW5	Rotary switch, one pole 12-pos set to 10-pos	Jameco 576501
SW2	Rotary switch, one pole 12-pos set to 11-pos	Jameco 576501
SW4	On-Off-On miniature toggle switch	Jameco 317308
Q1	2N6426	Mouser 610-2N6426
K1	Omron G5V-2-H1-DC5	Mouser 653-G5V-2-H1-DC5
LCD display	16x2 line LCD display	Jameco 2118598
Prototype boards		Jameco 2191411
Cabinet		Jameco 209358
Power supply	Triple output	Jameco 2210757
BNC panel mount		Jameco 355178
Dual banana jack		Jameco 2144024
Knobs (7)		Jameco 264946

PARTS LIST



■ **FIGURE 4.**
Microcontroller
schematic.

(the system 10 MHz clock).

For example, if the input signal is 1 MHz, then using the traditional frequency counter model and a one second gate interval, the error will be ± 1 input count in one million – not bad. If the input signal is 100 Hz, the ± 1 count is one in 100 – not very good. If we instead set the gate interval to be exactly 100 input cycles (one second), *TIMER2* will read 10 million ± 1 count – excellent.

In my design, I automatically switch to a period measurement

the PIC that is toggled high/low to simulate a clock pulse. The PIC then sets port D as an output, pulses the Master Clear pin on U3 setting the RAM address at 00000000, and then places the RAM data on the buss.

The PIC then toggles the *Write Enable* on the RAM to store the location 00000000 function data. The PIC then toggles PICCKC to advance U3 to RAM location 00000001, and repeats the storing sequence until all RAM locations are filled.

Frequency Counter

The frequency counter generally follows the traditional model where a fixed gate interval (say, one second) is set using a PIC counter (*TIMER2*) running off the 10 MHz clock. A second PIC counter (*TIMER1* – RC0 input) counts the number of input cycles during that interval. With a one second gate, the number of input cycles is equal to the input frequency (with a ± 1 count error).

This works very well if the input frequency is high because the ± 1 count error can be ignored. However, to accurately measure a low input frequency, the gate interval must either be increased, or the counter can be set to measure the period of the input signal.

With the latter technique, the roles of *TIMER1* and *TIMER2* are sort of reversed. *TIMER1* still sets the gate interval and still counts the input cycles, but it's programmed to count a specific whole number of input cycles so there is no ± 1 error here. Meanwhile, *TIMER2* counts the 10 MHz system clock during this gate interval; it has the ± 1 count error but the error is one in 10 million

when the input frequency is less than 150,000 Hz. I do this by making a preliminary measurement of the input frequency using a 0.1 second gate interval. If the frequency is greater than 150,000 Hz, I do a one second final frequency measurement. Otherwise, I switch to a period measurement. With this technique, I get frequency measurements accurate to five/six digits every 1.1 seconds over the entire frequency range of 10 Hz to over 10 MHz.

I want the period measurement to last for about one second. From the .1 second frequency trial measurement, we get an approximate value for the input frequency – say, 99 Hz. This means the input frequency is between 98 and 100 Hz (the ± 1 count error). Assume the input frequency is exactly 99.56 Hz. I set *TIMER1* to 99 (for the approximately one second gate time) and get the total number of 10 MHz clock pulses in 99 pulses of the input signal.

Since the input frequency is actually 99.56 Hz and it has a period of .0100442 seconds, the total gate time will be $99 \times .0100442$ or .994376 seconds.

Meanwhile, *TIMER2* is counting the 10 MHz clock (having a period of .0000001 seconds) and will end up with .994376/.0000001 or 9,943,760 clock pulses for 99 input cycles. So, the actual period of the input signal is $(9,943,760 \times .0000001)/99$ or .01004442 seconds, or 99.56 Hz – perfect!

There are many designs for a frequency counter input amplifier available on the Internet (also check out the design I used in the above referenced article). For this project, I simply use an available inverter (U8f) which is used for the high speed oscillator. To prevent any

interference, I turn the oscillator off in the frequency counter mode.

U2 is a 74F153 dual 4-to-1 multiplexer that is controlled by the PIC via the Select (SA, SB) lines as follows:

- 00.** Used in normal frequency counter mode (SIGIN from input amplifier).
- 01.** Used in frequency counter mode (SIGIN from input amplifier divided by four).
- 10.** Used in PFG mode with clock from high speed oscillator (HSCLK).
- 11.** Used in PFG mode with clock directly from the PIC (PICCLC).

U2a sets the clock input to the eight-bit counter (U3) to one of the following inputs:

- 00.** Frequency counter input (SIGIN from input amplifier).
- 01.** Frequency counter input (SIGIN from input amplifier).
- 10.** High speed clock (HSCLK).
- 11.** Clock from PIC (PICCLC).

U2b sets the input to the frequency input pin on the PIC (RC0) to one of the following inputs:

- 00.** The clock pin on the eight-bit counter (U3-CP).
- 01.** The second stage output on the eight-bit counter (U3 - I/O1).
- 10.** The clock pin on the eight-bit counter (U3-CP).
- 11.** The clock pin on the eight-bit counter (U3-CP).

PIC18F452 Microcontroller

I chose the PIC18F452 (see **Figure 4**) for this project for several reasons.

Its internal clock operates at 40 MHz, giving a 100 nanosecond instruction time. This is important when the frequency counter is operating in the period measurement mode or when you want to implement the PFG directly from the PIC (bypassing the RAM).

Because this PIC has a PLL that multiplies the crystal frequency by four, a 10 MHz crystal can be used. This reduces radiation that might interfere with the frequency counter input amplifier. The accuracy of the frequency counter is entirely dependent on the accuracy of the

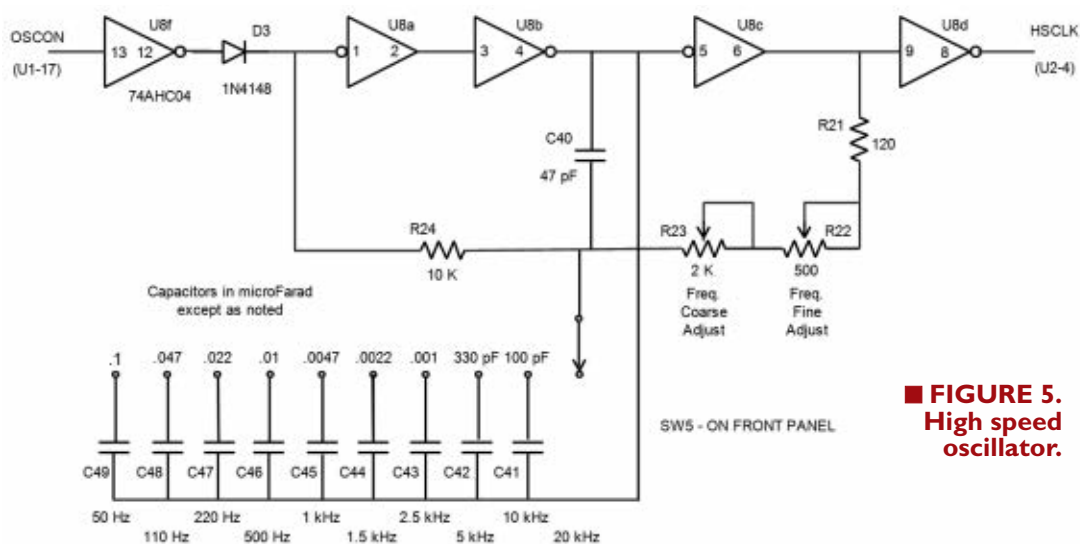


FIGURE 5. High speed oscillator.

crystal. A typical crystal has an accuracy at room temperature of about 100 PPM (parts per million); in other words, one in 10,000.

Many temperature controlled 10 MHz oscillators with one in 1,000,000 or better accuracy are available online at reasonable prices.

High Speed Oscillator

Figure 5 is the schematic for the high speed oscillator. It is basically a square wave relaxation oscillator using a 74AHC04 high speed hex inverter that will operate at a maximum frequency of about 15 MHz. The output (HSCLK) drives the eight-bit RAM clock input in the multiplexer '10' position (the normal PFG mode). At 15 MHz, the output of the PFG with 64-bit resolution is 15,000,000/256 or about 58 kHz.

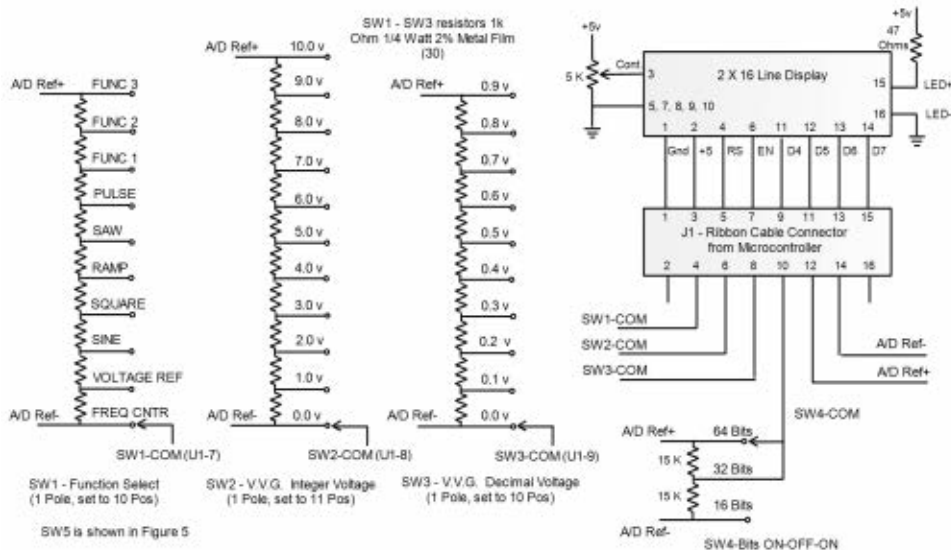
The output frequency of the oscillator is set by the band switch (SW5) and the two front panel potentiometers R22 and R23. The oscillator is turned off (via *SELINB*) when in the frequency counter mode.

Software

The software for this project was written in C using the microC development package, but any C compiler would work fine. The commented source code can be downloaded from the article link.

Front Panel

Photo 1 shows the layout of the front panel. On the left are the two voltage selectors for the VVR: one (SW2) for volts and the other (SW3) for tenth volts. The output of the VVR appears on the two banana jacks (VVR/PFG OUT).



■ **FIGURE 6. Front panel switch and display schematic.**

On the right of the front panel are the PFG controls:

- The Bits selection switch (SW4).
- The PFG amplitude potentiometer (**Figure 1**, R2).

popular solderless breadboards.

Construction drawings can be downloaded at the article link.

- The high speed oscillator frequency band switch (SW5)
- The Coarse and Fine frequency adjustment (**Figure 5**, R22 and R23).

In the center of the front panel is the Mode selector switch (SW1) used to choose between the frequency counter, VVR, and several other user preprogrammed PFG functions.

Construction

I mounted all the non-front panel components on two solderable prototype boards. The boards each have 1,258 points arranged in a grid much like other

WORLD'S MOST VERSATILE CIRCUIT BOARD HOLDERS

Model 324

Our Circuit Board Holders add versatility & precision to your DIY electronics project. Solder, assemble & organize with ease.

VISIT US ON

MONTHLY CONTEST

Visit us on Facebook® to post a photo of your creative PanaVise project for a chance to win a PanaVise prize package.

PANA VISE®
Innovative Holding Solutions

7540 Colbert Drive • Reno • Nevada 89511 | (800) 759-7335 | www.PanaVise.com

Model 201

Learn at Home!

Electronics Courses

Cleveland Institute of Electronics

Train at home to be a professional electronics or computer technician! Learn with hands-on labs, instructor support, online exams, videos and instructor led chat rooms.

FREE course catalog www.cie-wc.edu or (800) 243-6446

- **NEW!** Robotics Automation Lab
- **NEW!** Computer Security Specialist
- Industrial Electronics with PLC
- Broadcast Engineering
- Electronics with FCC
- PC Troubleshooting
- Electronics Troubleshooting
- Networking

Visit www.cie-wc.edu and start today!

CIE: 1776 E. 17th St., Cleveland, OH 44114 • (800) 243-6446 • Registration 70-11-0002H

Parts List

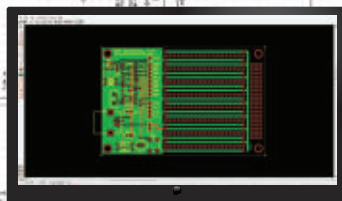
With the few exceptions given in the **Parts List**, all resistors are 1/4 watt, 2% metal film. I bought one of the resistor assortments listed on eBay, and now have enough resistors for many projects.

All 0.01, 0.1 bypass capacitors are ceramic disc and all capacitors in the microfarad range are tantalum. These can be purchased from Jameco or Mouser. The other capacitors are shown in the **Parts List**.

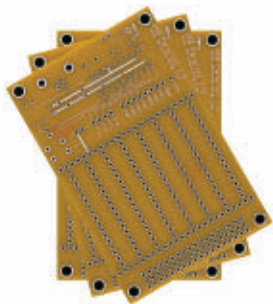
I hope you find this unit as convenient and helpful as I do. **NV**

FREE

PCB Layout Software &
PCB Schematic Software



expresspcb.com

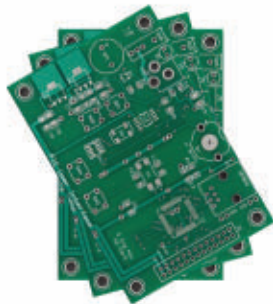


3 PCBs

\$41 + shipping

2-layer
MiniBoard

(Shipped in 1 business day)



3 PCBs

\$61 + shipping

2-layer
MiniBoardPro

(Shipped in 2 business days)



3 PCBs

\$81 + shipping

4-layer
MiniBoardPro

(Shipped in 3 business days)

- 1** **DOWNLOAD** our free CAD software
- 2** **DESIGN** your 2 or 4-layer PCB
- 3** **CHECK** your design with xCHECK DRC
- 4** **SEND** us your design with just a click
- 5** **RECEIVE** top quality boards in just days

Now Offering
**GROUND
SHIPPING**

- Continental US Customers -

LIMITED TIME PROMOTION | www.expresspcb.com

PoLabs

For more information or
software download please visit



www.poscope.com

PoKeys
Connect, Control



- USB
- Ethernet
- Web server
- Modbus
- CNC (Mach3/4)
- IO
- PWM
- Encoders
- LCD
- Analog inputs
- Compact PLC

Stepper motor drivers
Drive



- up to 256 microsteps
- 50 V / 6 A
- USB configuration
- Isolated
- up to 32 microsteps
- 30 V / 2.5 A

PoScope Mega1+
PoScope Mega50
Measure

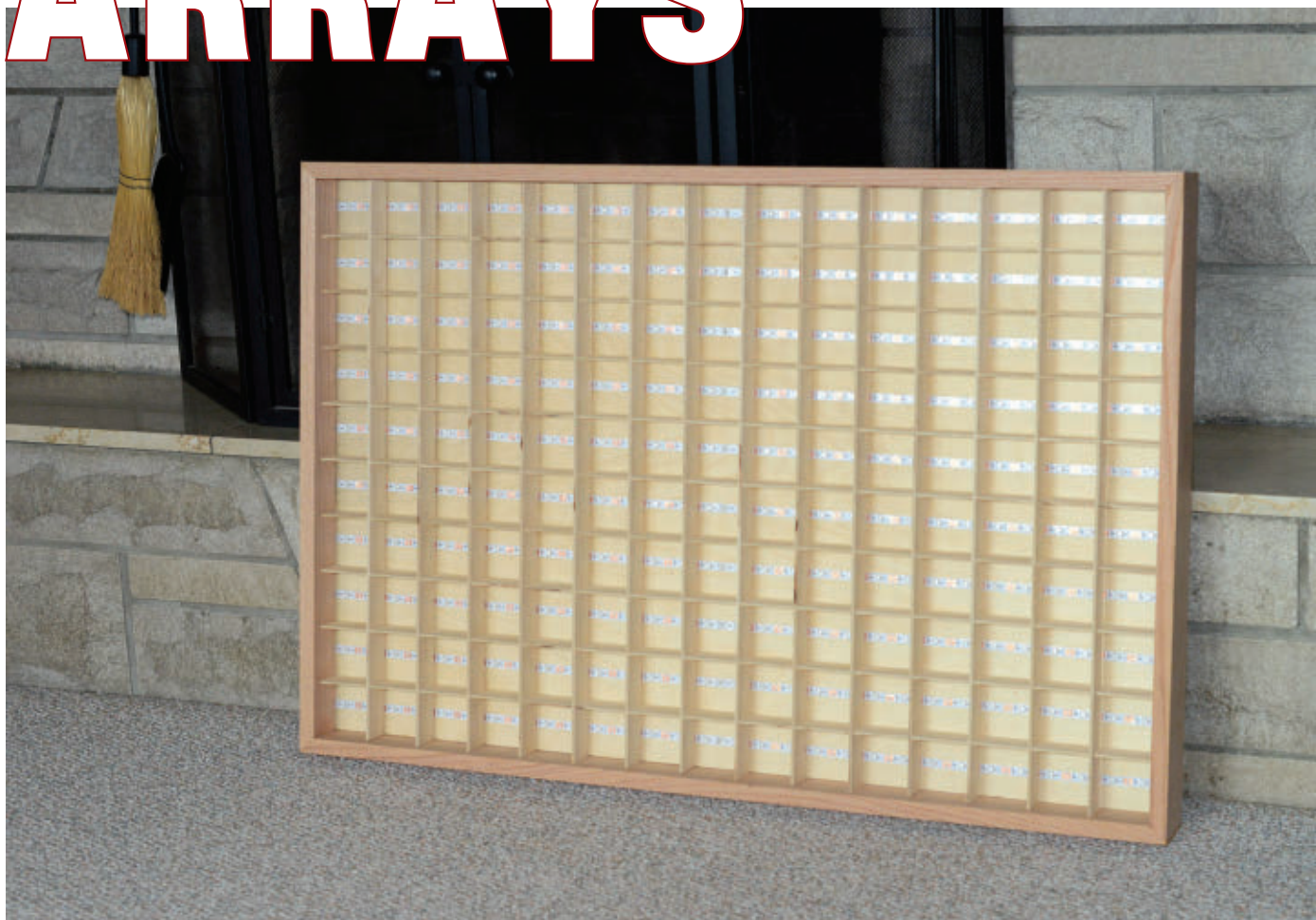


- up to 50MS/s
- resolution up to 12bit
- Lowest power consumption
- Smallest and lightest
- 7 in 1: Oscilloscope, FFT, X/Y, Recorder, Logic Analyzer, Protocol decoder, Signal generator

Using Programmable RGB LED Strips in ARRAYS

By Theron Wierenga

Post comments on this article and find any associated files and/or downloads at www.nutsvolts.com/magazine/article/November_2016_Programmable-RGB-LED-Strips-in-Arrays.



For some time, I have been thinking about building an LED table. Several iterations of table size, array size, type and number of LEDs, mount for the LEDs, circuits to drive the LEDs, and what exactly the array of LEDs would do has been considered. What I finally decided on was to use the popular RGB LED strips that are programmable. These strips are really easy to use and each LED can be programmed with millions of colors. Plus, they do not have to be continually scanned and refreshed. You select an LED and its color — or multiple LEDs — and then simply turn it on. These strips come in different densities (typically 30, 60, and 144 LEDs per meter) and are available from several vendors.

There is one caveat when using these RGB LED strips. When using solid colors like red, green, or blue, the color looks great. However, when you try to create a brown or lime green, for example, you will find that you have to tinker with the RGB values to get what you want. These strips do not translate the standard RGB triples as accurately as an LCD or LED monitor on a computer.

Adafruit is one supplier of these RGB LED strips, which they call Neopixels. They have some very good tutorials on how to practically use and program them. Go to <https://learn.adafruit.com/adafruit-neopixel-uberguide/overview> for details. Adafruit's Neopixel library is excellent and I use it for this project.

Let's start with a simple example of creating a half meter square array using strips with 30 LEDs per meter. Our array will be 15 LEDs square for a total of 225 LEDs. We will cut out strips with 15 LEDs and glue the strips down with a spacing of 3.33 centimeters (about 1-5/16 inch) between strips. Looking at the array in **Figure 1** and starting at the bottom right corner, the end of each strip needs to be wired to the start of the strip above at its left end. Looking at this as an X by Y array (X width and Y height), we can use the formula:

$$\text{num} = y * 15 + x$$

to locate the LED to program. Note that the value 15 is the height of our array. Using the Adafruit Neopixel library, we would use:

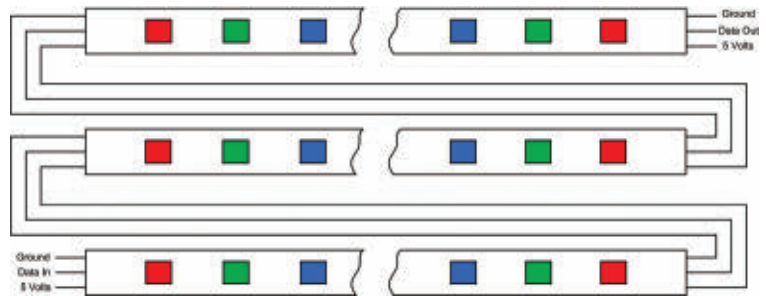
```
strip.setPixelColor(num, red, green, blue);
// sets the color, using values 0 - 255 for
// each color.
strip.Show();
// this turns on the LEDs you have programmed
// with colors.
```

Here's a complete program to draw a blue diagonal from the bottom left corner to the top right corner of our 15 x 15 array:

It's possible to mount the cut strips differently with every other strip going right to left. This will reduce the length of the wires needed between strips, but does require some additional software changes. Using our 15 x 15 array example, we would calculate num with this fragment:

```
if (y%2 > 0)
{
    num = y * 15 + x;
}
else
{
    num = y * 15 + (14 - x)
}
```

If you want to group RGB LEDs together (as in the chess board example), these equations get more involved.



■ **FIGURE 1. Wiring the strips together.**

```
#include <Adafruit_NeoPixel.h>
#include <avr/power.h>

// When we setup the NeoPixel library, we tell
// it how many pixels, and which pin to
// use to send signals. In our case 15 x 15
// = 225 pixels and we will use pin D6.
// Check the NeoPixel library for using the
// default NEO_GRB + NEO_KHZ800 value,
// which I found usually works just fine.

Adafruit_NeoPixel pixels =
Adafruit_NeoPixel(225, 6, NEO_GRB + NEO_KHZ800);

void setup()
{
    pixels.begin();
    // This initializes the NeoPixel library.
    pixels.setBrightness(32);
    // This sets the brightness of the LEDs
}

void loop()
{
    for(int i=0;i<15; i++)
    {
        int num = i * 15 + i
        // pixels.Color takes RGB values, from 0,0,0
        // up to 255,255,255
        pixels.setPixelColor(num,
        pixels.Color(0,0,128));
        // Moderately bright blue color.
    }
    pixels.show();
    // This sends the updated pixel color to
    // the hardware.
    delay(1000);
    // Delay for 1 second.
}
```

Here's another example. Create an array of 16 x 16 LEDs which will represent either a chess or checker board (an 8 x 8 array) using blocks of four LEDs for each square. Here are the four formulas to select the LEDs to turn on for a square at (x,y):

```
num1 = y * 32 + x
num2 = y * 32 + x + 1
num3 = y * 32 + 32 + x
num4 = y * 32 + 32 + x + 1
```

My Current Project

For my project, I made an array 30 LEDs wide by 10

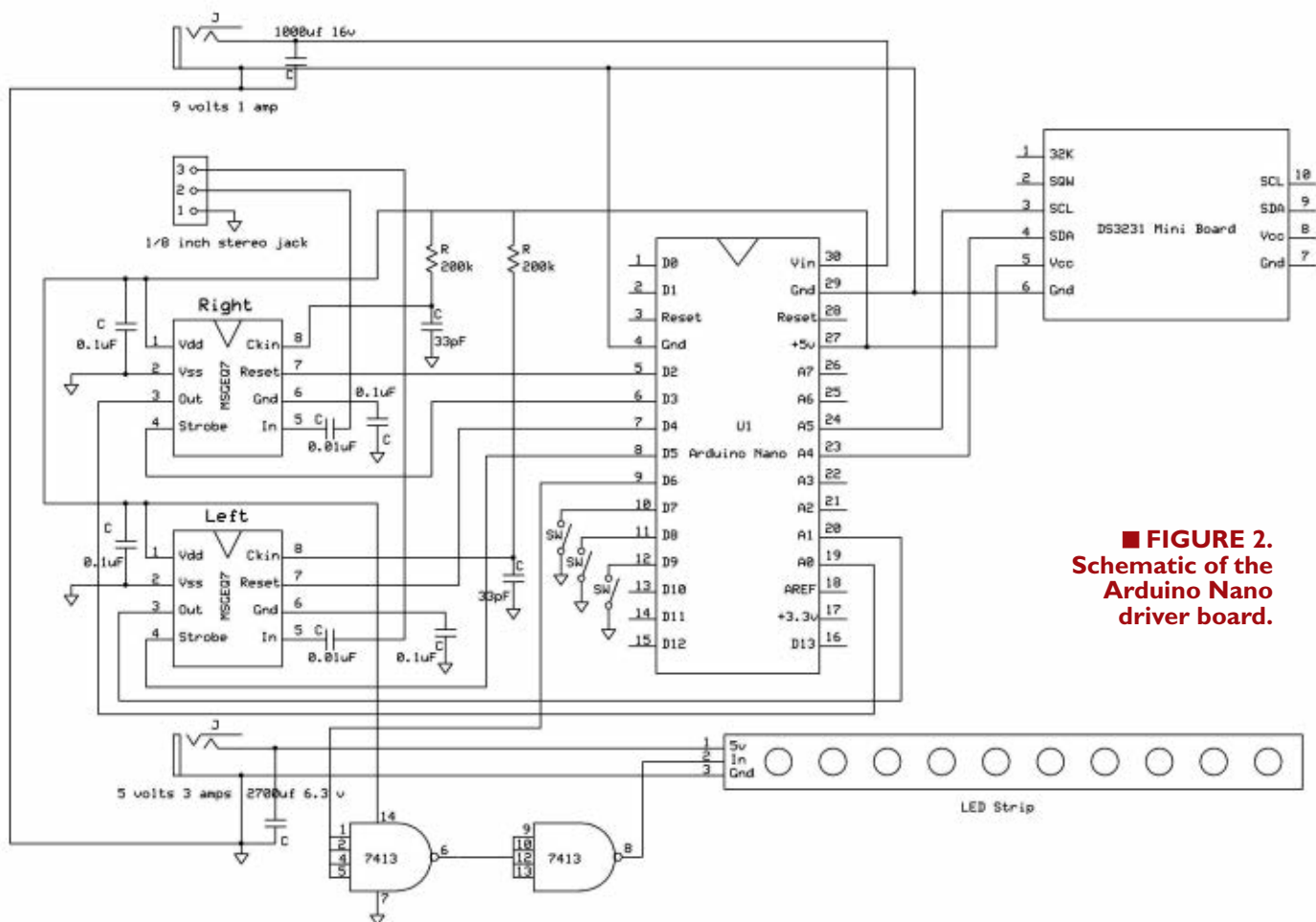


FIGURE 2.
Schematic of the
Arduino Nano
driver board.

high. I want two LEDs horizontally to represent each square in my 15 x 10 array. My formula for this 15 x 10 array then becomes $\text{num} = y * 30 + x * 2$. For the second LED, you just add one to num.

Here is what I did. I purchased two five meter long rolls of RGB LED strips. I chose the type that has 30 LEDs per meter with each LED 3.33 centimeters apart. I chose to create a 10 high by 15 wide array on my table top.

I should point out that this doesn't need to be a table; it could also be hung on a wall or mounted in other ways. I'm using two LEDs for each of my 10 x 15 squares, so that's $2 \times 3.33 \text{ cm} = 6.67 \text{ cm}$ or almost exactly 2.625 inches square for each cell.

I cut my LED strips into one meter lengths and used the adhesive strip on the back to mount them on a sheet of 1/4 inch plywood (keeping the rows 6.67 cm or 2.625 inches apart). I built an array of 2.625 inch square boxes using 1/8 inch plywood strips with 1/8 inch wide slits, put together in an interlocking grid. This grid is placed on top of the LED strips and serves to isolate the colors in each square. A sheet of glass or plastic is placed on the top of the box grid. If this sheet is not clear but diffuses the LED light, it will help even

ITEM

Two Five Meter RGB LED Strips, 30 LEDs per meter
 Arduino Nano
 Two MSGE07 Graphic Equalizer ICs
 Mini Board with DS3231 Real Time Clock and CR2032 Battery in Holder
 7413 Dual Four-input NAND Schmitt trigger IC
 Two 5.5 mm Power Jacks
 Mini Stereo Input Jack, 1/8 in
 Three Mini SPDT Toggle Switches
 158 x 90 x 60 mm Plastic Project Box
 Breadboard 3.75 x 2.875 in
 Four 0.1 µF Capacitors, 6.3 or higher voltage rating
 Two 0.01 µF Capacitors 6.3 or higher voltage rating
 Two 33 pF Capacitors, 10 volt voltage rating
 Two 200,000 ohm Resistors, 1/4 watt
 2700 µF Electrolytic Capacitor, 6.3 voltage rating
 1,000 µF Electrolytic Capacitor, 16 volt rating
 Nine volt/one amp Power Supply, compatible with 5.5 mm power jacks
 Five volt/three amp Power Supply, compatible with 5.5 mm power jacks
 Miscellaneous hookup wire, solder, header pins with sockets, screws, nuts, 1/4 and 1/8 inch plywood, 3/4 x 1 inch wood strips, and wood glue.

PARTS LIST

out the color of each box. The technique I used here will work for any size LED array you wish to make, limited only by the number of LEDs you can drive. I am using 300 LEDs in my project:

- $300 \text{ LEDs} \times 20 \text{ mA} \div 1,000 = 6.0 \text{ amps minimum}$, with all LEDs on with just one red, green, or blue color
- $300 \text{ LEDs} \times 60 \text{ mA} \div 1,000 = 18.0 \text{ amps minimum}$, with all LEDs on which produces white color

Realistically, you are probably not going to hit these high levels of current for any length of time, but they need to be kept in mind. In addition, a very long strip will begin to show a voltage drop at the far end from the power supply and the color of the LEDs will begin to be affected. The current use can also be reduced by using the command `pixels.setBrightness(32);` in the setup block. The brightness value can be between zero and 255. Because I will not be illuminating all the LEDs at one time, I chose a five volt/three amp wall wart to power my strip.

An Arduino Nano was used to drive the LED RGB array. Along with the minimum amperage issue, driving the single data in line from the Nano to the strip is also a concern. I found that after about a foot of length, this line can give problems. An Internet search uncovered the mention of using line drivers between the Arduino and the strip with success. I found that using a 7400 series Schmidt trigger also worked. I had a 7413 dual four input NAND Schmidt trigger IC in my parts bin and used this. The two NANDs were wired in series to produce a non-inverting output.

Two additional circuits were added to my design. The first was a mini board with a DS3231 real time clock and the second was a pair of MSGEQ7 seven band graphic equalizer ICs. This is the right amount of LEDs to display a four-digit clock. The pair of MSGEQ7 circuits was used to create a two-channel graphic equalizer display for

stereo output on the table.

I have built a number of different digital clocks and have always used the DS3231. The ± 2 PPM accuracy of this circuit means your clock will only be off about a minute per year. Recently, some small mini boards have appeared on the market with the DS3231 real time clock and the necessary CR2032 coin battery in a holder for backup. While the DS3234 has an SPI interface, the DS3231 has an I²C interface, making it a little easier to use

HF-AUTO High-Power Analog Autotuner



“ My HF-AUTO is a superb piece of engineering magic
W9OWO Dec 2012

“ Beyond awesome I will never sell this fast, accurate, reliable
W2PP Sept 2015

“ Of the 30+ tuners I have owned ... the HF-AUTO is the best
K1ESE Feb 2012

The HF-AUTO is a microprocessor controlled fully automatic stand-alone tuner with a power rating of 5 Watts to 1800 Watts that will work with any transmitter built from the 1940s to the present. HF Bands: 160m to 6m. Three antenna outputs: SO239 coax. Dimensions: 12.5" W x 6.5" H x 16.5" D. Weight: 25 lbs. (11.4 kg).



HF-AUTO-R Remote unit
Operate from 500' distance

Palstar product reviews from actual Hams as found on eHam.net

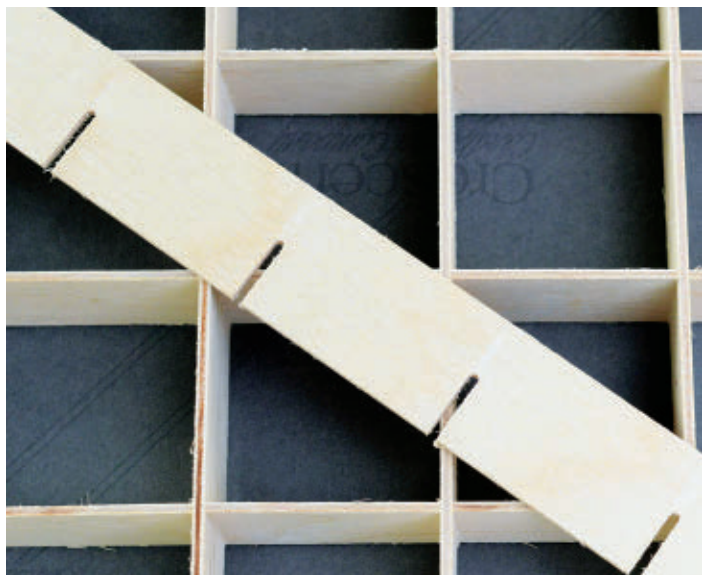
PALSTAR

Customer Support: 1-800-773-7931
Fax: 937-773-8003

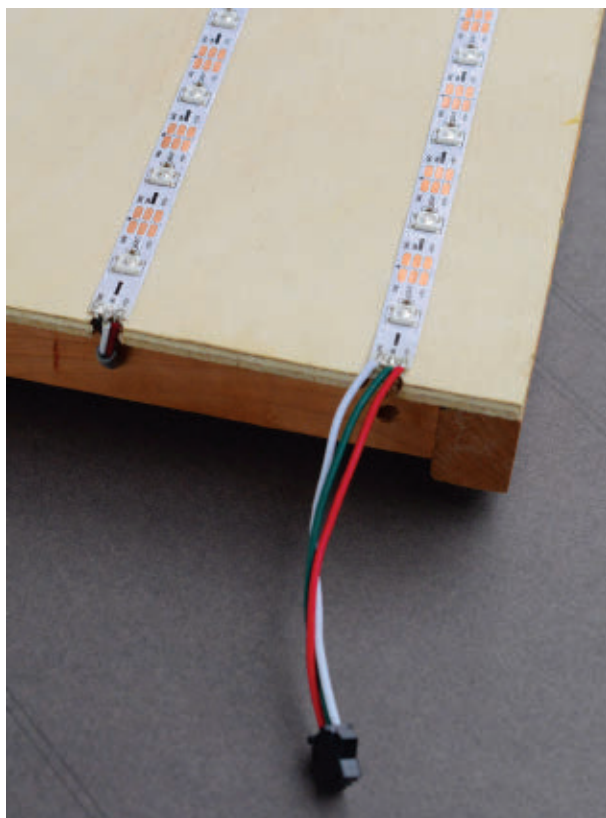
www.palstar.com

with fewer connections. The RTCLib library makes using this mini board very easy. RTCLib can be found at GitHub at <https://github.com/adafruit/RTCLib>.

The MSGEQ7 circuits are a little more involved, but with the addition of four capacitors and a resistor, you



■ **FIGURE 4.** Box array, with each box fitting over two LEDs to isolate the colors.



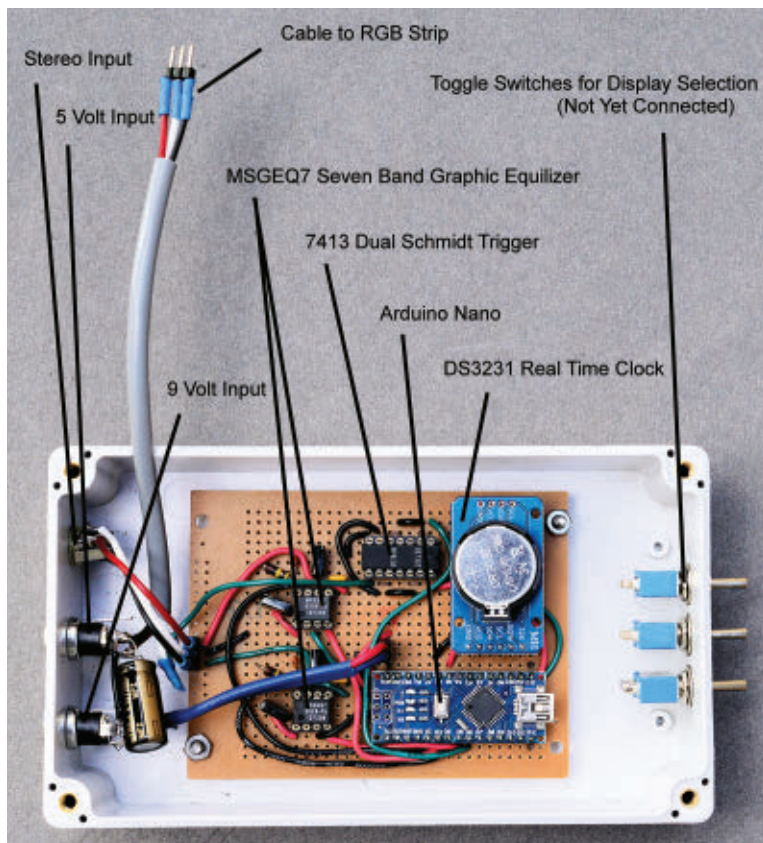
■ **FIGURE 3.** Edge of the plywood board the RGB LED strips are mounted on, showing the wiring.

have a single-channel graphic equalizer display; with two, you get a display for stereo output. Using this circuit is fairly straightforward. After sending a reset pulse, the strobe line is pulsed low and the output is then read using an *analogRead()* command. This is repeated seven times, giving numeric values from 0-1023 for frequency channels with peaks at 63 Hz, 160 Hz, 400 Hz, 1 kHz, 2.5 kHz, 6.25 kHz, and 16 kHz. The complete schematic for my design is shown in **Figure 2**.

The Build

I mounted my RGB LED strips on a 1/4 inch piece of good quality plywood. Reinforcement pieces of 3/4 x 1 inch wood were glued around the edges of the plywood in addition to two added ribs to give the plywood rigidity. Small notches were cut in the edges of the plywood to give the connecting wires a path to the underside of the plywood. A number of holes must also be drilled in the ribs to allow the connecting wires to pass through.

The driver circuit was built on a breadboard with point-to-point wiring. This board was enclosed in a 158 x 90 x 60 mm plastic project box. The sides of the box were pierced with holes for mounting two 5.5 mm jacks for nine volt/one amp and five volt/three amp power supplies (the wall wart type). The five volt power supply is only



■ **FIGURE 5.** Inside the project box, showing component placement.

used to power the RGB LED strip. There is also a 1/8 inch mini jack for a stereo input plug and three small toggle switches to determine which function the driver board will perform. A short three-wire cable was brought out to connect to the RGB LED strip.

Software

The RGB LED table was designed primarily to do two things: display a stereo graphic equalizer and a four-digit clock. Here is the *freq()* function to display the graphic equalizer:

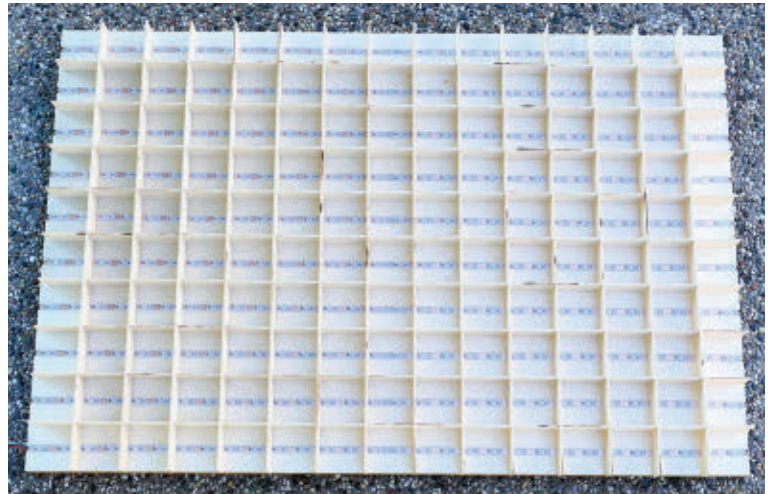
```
// Read the two stereo MSGEQ7 audio
// channels and display a graphic
// equalizer
void freq()
{
    int num;
    // Read frequencies and display
    // Left channel
    // Do a reset
    digitalWrite(resetPinLeft, HIGH);
    digitalWrite(resetPinLeft, LOW);

    // Read the left seven frequency channels
    for (int i = 0; i < 7; i++)
    {
        // Set strobe low then read a value
        digitalWrite(strobePinLeft, LOW);
        // Allow output to settle
        delayMicroseconds(30);
        // Dividing by 103 reduces the 0 - 1023
        // output to 0 to 9
        spectrumValueLeft[i] = analogRead
            (analogPinLeft) / 103;
        // Return strobe to high, ready for
        // another read
        digitalWrite(strobePinLeft, HIGH);
    }

    // Right channel
    digitalWrite(resetPinRight, HIGH);
    digitalWrite(resetPinRight, LOW);

    // Read the right seven frequency
    channels
    for (int i = 0; i < 7; i++)
    {
        digitalWrite(strobePinRight, LOW);
        delayMicroseconds(30);
        spectrumValueRight[i] = analogRead
            (analogPinRight) / 103;
        digitalWrite(strobePinRight, HIGH);
        // Serial.print(spectrumValueRight[i]);
    }

    // Now display the results
    // Left channel
    for (int x = 0; x < 7; x++)
    {
        // seven channels
        {
            for (int y = 0; y < 10; y++)
            {
                // representing frequency strength
                num = y * 30 + x * 2;
                if (spectrumValueLeft[x] >= y)
                // add a color to the bar height
                {
                    pixels.setPixelColor(num, colors
```



■ FIGURE 6. The square boxes mounted on top of the RGB LED board.



■ FIGURE 7. Back side of the RGB LED board, showing the wiring between the strips.

```
        [x + 1]);
        pixels.setPixelColor(num + 1,
            colors[x + 1]);
    }
    else
    {
        // or fill in the top with no color
        pixels.setPixelColor(num, 0, 0, 0);
        pixels.setPixelColor(num + 1, 0, 0,
            0);
    }
}

// Right channel
for (int x = 0; x < 7; x++)
{
    for (int y = 0; y < 10; y++)
    {
        num = y * 30 + x * 2 + 16;
        if (spectrumValueRight[x] >= y)
        {
            pixels.setPixelColor(num, colors
                [x + 1]);
            pixels.setPixelColor(num + 1,
```



```

        colors[x + 1]);
    }
    else
    {
        pixels.setPixelColor(num, 0, 0, 0);
        pixels.setPixelColor(num + 1, 0, 0, 0);
    }
}
// After every LED is programmed turn them on
pixels.show();
}

```

These two functions read the DS3231 real time clock and display the results:

```

// Read and display the time
// hourmin set to HOUR for a 4 digit hour and
// minute clock or MIN for a 4 digit minute and
// second clock
void getTime(int hourmin)
{
    // Read the current time from DS3231 RTC
    DateTime now = rtc.now();
    int h = now.hour();
    if (h > 12) h = h - 12;
    // a 12 hour clock, remove for 24 hour
    int m = now.minute();
    int s = now.second();
    // now split into 10s and 1s digits
    int h10 = h / 10;
    int h1 = h % 10;
    int m10 = m / 10;
    int m1 = m % 10;
    int s10 = s / 10;
    int s1 = s % 10;

    // Now display the 4 digits
    switch (hourmin)
    {
        case HOUR: setLedNumbers(h10, 0);

```

When programming the DS3231 real time clock with the Arduino Nano, here's an important point not to miss. You program the DS3231 with the *rtc.adjust(DateTime(year, month, day, hour, minute, second))* function in the setup block of the program. This function includes the current time and date at your location. This will set the real time clock going in the DS3231; with the installed CR2032 backup battery, the clock will continue to run when power is removed.

Next, comment out the line with the *rtc.adjust(DateTime(year, month, day, hour, minute, second))* function and program the Nano again without it. Now if you remove power, the DS3231 will continue running; when you power-up the Nano without the *rtc.adjust(DateTime(year, month, day, hour, minute, second))* function, it will not overwrite the clock's current time.

Notice also that the five volt power for the DS3231 comes from the five volt output of the Nano. This allows you to plug in your USB cord to program the Nano, plus the DS3231 will be powered as well (which is convenient for debugging).

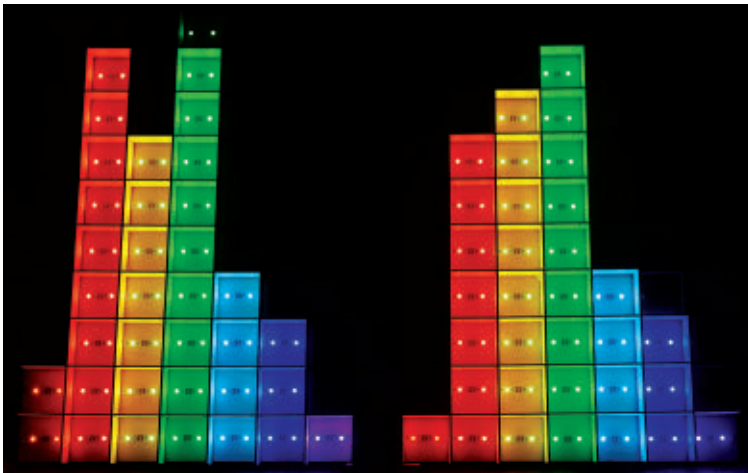
When setting the time for the DS3231, there appears to be some irregularities when setting the seconds value to 0. The minute value is off by one and the seconds is incorrect. When the initial seconds value is greater than 0, this doesn't appear to happen. There may be a bug in the *rtc.adjust* or *DateTime* portions of the code, so check your results after setting the time.

```

        setLedNumbers(h1, 1);
        setLedNumbers(m10, 2);
        setLedNumbers(m1, 3);
        break;
    case MIN: setLedNumbers(m10, 0);
              setLedNumbers(m1, 1);
              setLedNumbers(s10, 2);
              setLedNumbers(s1, 3);
              break;
    }
}

// Display a single digit
// place = 0 displays digit in UL
// place = 1 displays digit in UR
// place = 2 displays digit in LL
// place = 3 displays digit in LR
void setLedNumbers(int digit, int place)
{
    int num;
    byte p, q;
    // Set the color of each digit
    switch (place)
    {
        case 0: color = colors[ORANGE];
                break;
        case 1: color = colors[ORANGE];
                break;
        case 2: color = colors[BLUE];
                break;
        case 3: color = colors[BLUE];
                break;
    }
    for (int y = 0; y < 7; y++)
        // seven rows
    {
        p = 16;
        for(int x = 0; x < 5; x++)
            // five columns
        {
            q = Font1[digit][y] & p;
            // AND each bit in the 5x7 array
            switch (place)
            {
                case 0: // num = x * 30 + y * 2;
                // These alternate values just
                // switches display horizontally
                num = (9 - x) * 30 + (28 - (y * 2));
                break;
                case 1: // num = (x + 5) * 30 + y * 2;
                // 2;
                num = (4 - x) * 30 + (28 - (y * 2));
                break;
                case 2: // num = x * 30 + y * 2 +
                // 16;
                num = (9 - x) * 30 + (12 - (y * 2));
                break;
                case 3: // num = (x + 5) * 30 + y * 2 +
                // 2 + 16;
                num = (4 - x) * 30 + (12 - (y * 2));
                break;
            }
            // Does the value in the 5x7 array need
            // to be turned on?
            if (q > 0)
            {
                pixels.setPixelColor(num, color);
                pixels.setPixelColor(num + 1, color);
                Serial.print("*");
            }
            else
            {

```



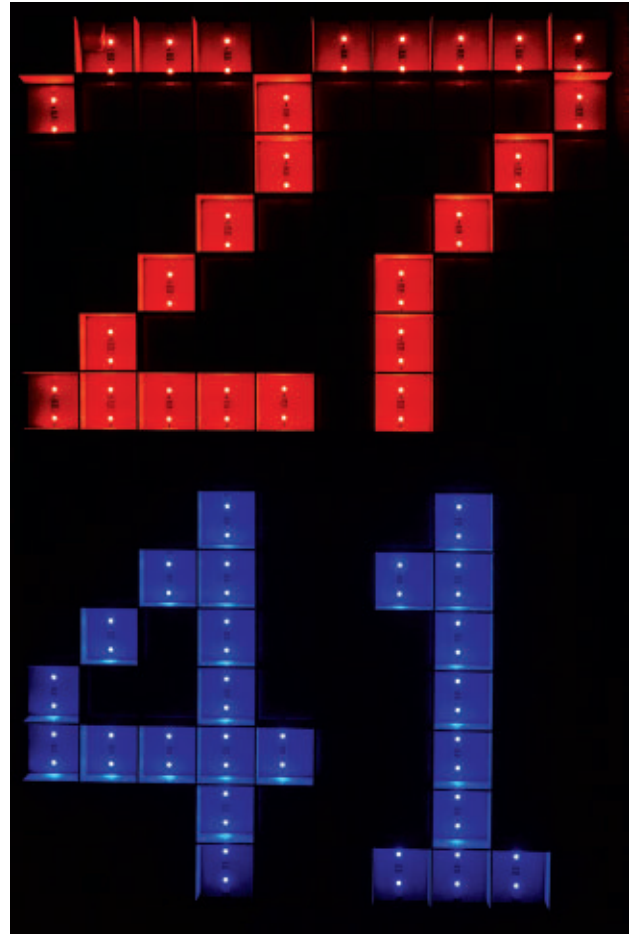
■ **FIGURE 8.** RGB LED table functioning as a stereo graphic equalizer.

```


pixels.setPixelColor(num, colors
[BLACK]);
pixels.setPixelColor(num + 1,
colors[BLACK]);
Serial.print(" ");
}
// Shift once to right for next bit
p = p >> 1;
}
// Now turn on the display
pixels.show();
}

```

The complete program is available at the article link. After the necessary values are initialized in the *setup()* function of the Arduino program, the main *loop()* function is executed, which simply reads the positions of the three toggle switches and uses a switch statement to decide what the RGB LED table will display. **NV**





■ **FIGURE 9.** The RGB LED table functioning as a large digital clock, displaying minutes and seconds.



**Electronix
EXPRESS**

Your Single Source of Electronic Equipment, Tools, and Components

<p>01DM820B Digital Multimeter Super Economy</p>  <p>DMM - Volts, current, Resistance, Transistor and Diode Test. Includes Leads</p> <p style="text-align: right;">\$7.50</p>	<p>01DMMY64 Digital Multimeter 32 Ranges</p>  <p>Including temperature, capacitance, frequency and diode/transistor testing</p> <p style="text-align: right;">\$29.95</p>	<p>01DMM9803R Bench DMM True RMS</p>  <p>Auto and manual range, RS232C standard interface. Runs on DC or AC</p> <p style="text-align: right;">\$139.95</p>	<p>Sweep Function Generator</p> <p>3MHz, 6 Waveform Functions, Int/Ext Counter, lin/log sweep</p> <p>Model FG-30 (no display) \$135</p> <p>Model FG-32 (5 digit display) \$195</p> 
<p>01DS1102E 100MHz Rigol Oscilloscope</p>  <p>Features: 1 million points of deep memory, FFT, record and replay, roll mode, alternate trigger mode, and adjustable trigger sensitivity</p> <p style="text-align: right;">\$399</p>	<p>01SDS1052DL Siglent 50MHz DSO</p>  <p>500MSa/s Sampling 32K Memory 6 Digit Frequency Counter USB Interface 7" Color TFT-LCD Screen</p> <p style="text-align: right;">\$279</p>	<p>DC Power Supplies</p> <p>Model HY3003 Variable Output, 0-30VDC, 0-3Amp. \$95</p> <p>Model HY3003-3 Two 0-30VDC, 0-3Amp Variable Outputs plus 5V 3A fixed. \$195</p> 	
<p>Parallax Board of Education Full Kit</p> <p>USB version with Stamp and USB Cable.</p> 		<p>BOE Full Kit Parallax 7.5V Power Supply</p> <p style="text-align: right;">\$98.90 \$8.95</p>	<p>0603ZD98 48W Soldering Station</p> <p>Temp. Controlled 310°F-840°F</p> <p style="text-align: right;">\$15.25</p> 

1 (800) 972-2225

| http://www.elexp.com

| contact@elexp.com

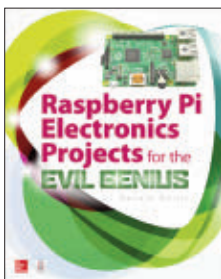
For complete product details, visit us online!!

The Nuts & Volts **WEBSTORE**

GREAT FOR DIYers!

Raspberry Pi Electronics Projects for the Evil Genius by Donald Norris

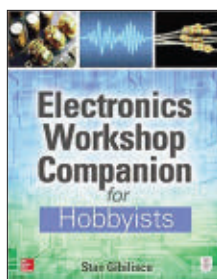
This fully illustrated TAB guide shows how to construct and program all kinds of fun and innovative gadgets with your Raspberry Pi. *Raspberry Pi Electronics Projects for the Evil Genius* features 10 complete projects that showcase cool RasPi applications in computing, communications, robotics, photography, and video. Each inexpensive project includes a detailed list of materials, sources for parts, schematics, and clear step-by-step assembly and programming instructions.



\$25.00

Electronics Workshop Companion for Hobbyists by Stan Gibilisco

In this practical guide, electronics expert Stan Gibilisco shows you, step by step, how to set up a home workshop so you can invent, design, build, test, and repair electronic circuits and gadgets. *Electronics Workshop Companion for Hobbyists* provides tips for constructing your workbench and stocking it with the tools, components, and test equipment you'll need. Clear illustrations and interesting do-it-yourself experiments are included throughout this hands-on resource.



\$25.00

Arduino: A Beginner's Guide To Programming Electronics by Chuck Hellebuyck

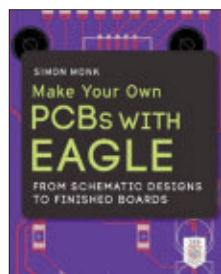
Arduino is one of the most popular microcontroller development boards available. It's more than just a module though, as the heart of Arduino is the software you run on your computer and use to create sketches (or programs) in the C programming language, with many pre-built functions. Chuck explains all this in very simple terms to help you get started programming your own electronic applications. His down-to-earth style of explaining technology makes it a great place to get started with programming electronics.



\$15.00

Make Your Own PCBs with EAGLE by Eric Kleinert

Featuring detailed illustrations and step-by-step instructions, *Make Your Own PCBs with EAGLE* leads you through the process of designing a schematic and transforming it into a PCB layout. You'll then move on to fabrication via the generation of standard Gerber files for submission to a PCB manufacturing service. This practical guide offers an accessible, logical way to learn EAGLE and start producing PCBs as quickly as possible.

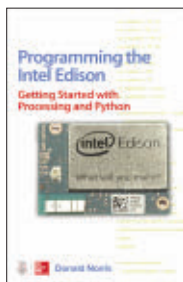


\$30.00

Programming the Intel Edison: Getting Started with Processing and Python by Donald Norris

Learn To Easily Create Robotic, IoT, and Wearable Electronic Gadgets!

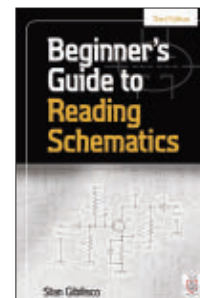
Discover how to set up components, connect your PC or Mac, build Python applications, and use USB, WiFi, and Bluetooth connections. Start-to-finish example projects include a motor controller, home temperature system, robotic car, and wearable hospital alert sensor.



\$20.00

Beginner's Guide to Reading Schematics, 3E by Stan Gibilisco

Navigate the roadmaps of simple electronic circuits and complex systems with help from an experienced engineer. With all-new art and demo circuits you can build, this hands-on, illustrated guide explains how to understand and create high-precision electronics diagrams. Find out how to identify parts and connections, decipher element ratings, and apply diagram-based information in your own projects.

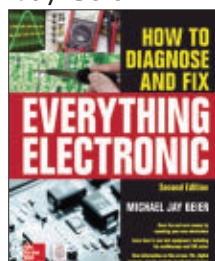


\$25.00

How to Diagnose and Fix Everything Electronic by Michael Jay Geier

A Fully Revised Guide to Electronics

Repair all kinds of electrical products, from modern digital gadgets to analog antiques, with help from this updated book. The Second Edition offers expert insights, case studies, and step-by-step instruction from a lifelong electronics guru. Discover how to assemble your workbench, use the latest test equipment, zero in on and replace dead components, and handle reassembly.



\$24.95

Programming PICs in Basic by Chuck Hellebuyck

If you wanted to learn how to program microcontrollers, then you've found the right book! Microchip PIC microcontrollers are being designed into electronics throughout the world and none is more popular than the eight-pin version. Now the home hobbyist can create projects with these little microcontrollers using a low cost development tool called the CHIPAXE system and the Basic software language. Chuck Hellebuyck introduces how to use this development setup to build useful projects with an eight-pin PIC12F683 microcontroller.



\$14.95

Programming Arduino Next Steps: Going Further with Sketches by Simon Monk

In this practical guide, electronics guru Simon Monk takes you under the hood of Arduino and reveals professional programming secrets. Also shows you how to use interrupts, manage memory, program for the Internet, maximize serial communications, perform digital signal processing, and much more. All of the 75+ example sketches featured in the book are available for download.



\$20.00

Order online @ store.nutsvolts.com
Or CALL 1-800-783-4624 today!

EDUCATIONAL

Beginner's Guide Educational Combo!

Price Reduced
Now Only \$219.95

An Arduino Workshop

Are you puzzled about the Arduino but finding it difficult to get all the pieces in one place?

Joe Pardue
SmileyMicros.com

Puzzled by the Arduino?

Based on *Nuts & Volts* Smiley's Workshop, this set gives you all the pieces you need!

Educational Combo \$124.95

We now have the fully updated Learning Labs 1, 2, & 3 in stock

Are you (or someone in your family) looking to get started in engineering, electronics, computers, or robotics? If so, this hands-on comprehensive Learning Lab series is for you! Each Lab includes hands-on exercises and training on practical electronics, components, and circuits. Complete details on all the Learning Labs are available in our website. Simply type "lab" into the search field at <http://store.nutsvolts.com>!

Now \$45.95

Now \$47.50

Now \$41.95

Educator Discounts Available!

CD-ROM SPECIAL

Nuts & Volts 12 CD-ROMs & Hat Special!

That's 144 issues. Complete with supporting code and media files.

Free Shipping!

Only \$249.95 or \$24.95 each.

The Nuts & Volts Pocket Ref

All the info you need at your fingertips!

This great little book is a concise all-purpose reference featuring hundreds of tables, maps, formulas, constants & conversions. AND it still fits in your shirt pocket!

Only \$12.95

Visit store.nutsvolts.com or call (800) 783-4624

Strings & Things

A frequent topic of discussion by newcomers on the Propeller forum is the use of strings. Spin doesn't have an explicit string type, so this can be a little confusing at first. Have no fear! Working with strings is very simple. Spin even has a few built-in tools to deal with them. Anything else we can code on our own, and it's well worth doing — especially for use in HMI's and user displays.

As with many other languages, a string in Spin is simply an array of characters that is terminated by a null character (zero); this termination is critical. Let's say, for example, that we want to work with strings that are up to 15 characters long. At a minimum, we'll need to define an array that holds 16 bytes (15 characters plus the terminator):

```
var

byte  buffer[16]
```

Okay, no problem. How then, do we pass this string to a method to be worked on? All parameters in Spin are longs, which means passing a byte, word, or long is no problem. Again, what do we do about strings (or other arrays for that matter)? The key is *passing by reference* — a term you may have encountered in a programming book. The other side of that is *passing by value*:

```
myVar := 100
do_something(myVar)
```

This is an example of *passing by value*. What's happening here is that the value of *myVar* is being copied onto the stack before the call to **do_something()** where it gets used. In this case, the original contents of *myVar* are not changed.

With no explicit string type, we are forced to pass strings *by reference*; this means that we're passing the address of the string, not the contents of the string. You'll remember that the **@** operator is used to get the address of a variable.

Most objects that send the contents of a string to an output device have a method called **.str()** — you'll find this in **FullDuplexSerial** and similar objects. To send the contents of the buffer to a terminal for display, we could do this:

```
term.str(@buffer)
```

Let's have a look at the method as there are a couple things worth pointing out:

```
pub str(p_str)
  repeat strsize(p_str)
    tx(byte[p_str++])
```

This is from my modification of **FullDuplexSerial**. The parameter is specifically named to remind me that this is a pointer (the address of). Inside the method, we find **strsize()** — one of the built-in functions for working with strings. The function returns the number of characters in the string, not counting the terminating zero. If we had to code this manually, it would look like this:

```
pub str_size(p_str) | len

len := 0
repeat
  if (byte[p_str++] == 0)
    return len
  else
    len += 1
```

As you can see, the method iterates through and counts the characters in the string until it finds the terminator. I'm emphasizing this because the absence of a terminator can result in run-away output to a device — sometimes it's not pretty, believe me.

The most frequent use of strings is for messages in a program. In Spin, we predefine strings in a **DAT** table:

```
dat

Name      byte    "Jon McPhalen", 0
Phone     byte    "818-555-1212", 0
```

From time to time, you will run into inline strings:

```
term.str(string("Jon McPhalen"))
```

The **string()** function creates a string that is embedded in place and returns its address. Other than debugging statements, I tend to avoid inline strings because it means hunting them down if I want to modify them. By grouping everything in a single location (**DAT** table), the strings for a

program are easier to maintain, and can be re-used from multiple locations within an object.

I've written a lot of programs that need to retrieve the Nth string from a built-in list. Let's say, for example, that we have an index and want to print the string associated with it. One way is to create a list of strings that are all of the same length. We do this by padding the end of each string with zeroes so all strings match the length of the longest:

dat

Names	byte	"Jon",	0[3]
	byte	"Lynda",	0
	byte	"Farah",	0
	byte	"Ken",	0[3]
	byte	"Rick",	0[2]
	byte	"Matt",	0[2]
	byte	"Dirk",	0[2]

As you can see, Lynda and Farah have the longest names — each of which occupies six bytes in memory. The others are shorter and have to be padded to six bytes. We can access a specific name in the list by using the index and a little math:

```
term.str(@Names + (idx * 6))
```

The addition advances the pointer into the table as required by the index. There are two downsides to this strategy: 1) We chew up a few extra bytes with padding; and 2) As soon as we add a longer name to the list, we have to edit everything. I have a wonderful friend named McKenzie who represents me in my acting life. If I want to add her name to the list, I have to pad everything to nine bytes, and then change the multiplier used with the index.

Spin has a mechanism that can help, and I use this technique if the list isn't too long. For example, we could have this list of names for days of the week:

dat

Day0	byte	"Sunday",	0
Day1	byte	"Monday",	0
Day2	byte	"Tuesday",	0
Day3	byte	"Wednesday",	0
Day4	byte	"Thursday",	0
Day5	byte	"Friday",	0
Day6	byte	"Saturday",	0

With names for each string, we can use @, but for an application that reads the day number (0..6) from an RTC (real time clock), we need to find the correct string with that index. To do that, we add a second table which are

addresses of the original strings. Note that these are defined as words because we don't know where they're being placed in the 32K program space:

Weekday	word	@Day0, @Day1, @Day2, @Day3
	word	@Day4, @Day5, @Day6

When the code is compiled, the address of each string — within its object — will be stored in the table. To correctly use this table requires the @@ operator; this adds the object offset to the compile-time addresses in the table so that a string-related method will work. Here's how it's used:

```
today := rtc.get_day  
term.str(@@Weekday[today])
```

where *today* is the index into the *Weekday* table. Again, this works great with small tables, but can be tedious to set up for long tables, or in a case where the table is a variable length (for example, a table read in from a file). Before I get to that, let me show you another approach to the table above. In my strings library, I have a method called **.str_ptr()** which will return the address of the Nth string in a list:

```
pub str_ptr(idx, p_list)  
  
repeat idx  
  p_list += strsize(p_list) + 1  
  
return p_list
```

To use this, we'd change the program to:

```
term.str(str_ptr(today, @Day0))
```

This is easier to set up, but a little slower as it has to iterate through all the strings in the list until it finds the target. If you're wondering about the + 1 in the code, remember: We need this to skip over the terminating 0 as that is not included in **strsize()**.

Once we get used to using pointers, they open up a lot of interesting possibilities. One of my best days as a Toro employee was receiving (with a colleague) the **Circle of Excellence** award for developing the first multilingual golf course irrigation control system for the company. I was married to a foreigner, and traveled the world a fair amount, so I was very sensitive to how those who don't speak English as a first language were sometimes frustrated by English-only products from the US. Okay, then, how can we use what we know to create a multilingual Propeller app?

Step 1 is to create the strings tables — we'll keep it

simple for the sake of demonstrating things. The product I just referred to supported six languages:

```
dat

English      byte    "Hello", 0
              byte    "Goodbye", 0

French       byte    "Bonjour", 0
              byte    "Au Revoir", 0

German       byte    "Hallo", 0
              byte    "Auf Wiedersehen"

Italian      byte    "Ciao", 0
              byte    "Arrivederci", 0

Japanese     byte    "Kon'nichiwa", 0
              byte    "Sayoonara", 0

Spanish      byte    "Hola", 0
              byte    "Adiós", 0
```

To make the program easy to read, let's set up constants for the supported languages:

```
con

L_ENGLISH    = 0
L_FRENCH     = 1
L_GERMAN     = 2
L_ITALIAN    = 3
L_JAPANESE   = 4
L_SPANISH    = 5
```

In such a program, we'd have a control variable that points to the correct language table; it could be initialized like this:

```
case language
  L_ENGLISH : p_msgs := @English
  L_FRENCH  : p_msgs := @French
  L_GERMAN  : p_msgs := @German
  L_ITALIAN : p_msgs := @Italian
  L_JAPANESE : p_msgs := @Japanese
  L_SPANISH : p_msgs := @Spanish
  other     : p_msgs := @English
```

Once *p_msgs* is set to our language choice, we can put it to use:

```
display(str.str_ptr(MSG_HELLO, p_msgs))
```

Just as there are many roads that lead to Rome, there are many strategies to solve a problem, and I seem to find

myself combining disparate strategies for the best solution. A few months ago, my friend, Dirk – whom I met because he reads this column – asked for some help on a project he wanted to take to Burning Man. Dirk is one of the nicest people I've ever met, and does some really amazing things for his clients – sometimes using bits of my code. So, I always enjoy pitching in a bit when he needs a code assist.

His specific need was to randomly play about 100 WAV files without repeating until all had played. The only way to do this – without forcing a specific naming convention on the files (something he didn't want to do) – was to take an inventory of the WAV files on the SD card and create a list in RAM.

The blank list is created in a **DAT** table:

```
dat

WavList      byte    0[8 * MAX_FILES]
FileExt      byte    ".WAV", 0
```

I use FSRW for file access, and it can only read 8.3 file names from the SD card. Even if we use a long file name on the SD card, the operating system creates a short version. We might see:

`whole_lotta_love.wav`

when looking at the file in Windows, but FSRW will read it as:

`whole_~1.wav`

I know what you're wondering: What if there is another file that has "whole_" as the first characters. No worries! The operating system will change the number that follows the tilde.

Since we're only playing WAV files, we don't need to store the extension. The string called *FileExt* is used to filter WAV files from those on the SD, and we can add this to the selected file name before calling the WAV player object.

First things first: We need to read the directory of the SD card and save the names of the WAV files into the RAM array:

```
pub inventory_wav_files | check, ofs

wav.open_dir
filecount := 0

repeat MAX_FILES
  check := wav.next_file(@filename)
  if (check == 0)
    str.ucstr(@filename)
```

```

    ofs := str.instr(@filename, @FileExt)
    if (ofs > 0)
    bytefill(@filename+ofs 0, 4)
    ofs := filecount << 3
    bytemove(@WaveList+ofs, @filename, 8)
    filecount += 1
    else
    quit

wav.close_file

```

What's happening here is that we're reading the file names using the **.next_file()** method. If there is a valid file, the 8.3 name will be moved into the array called *filename*, and *check* will be zero (this is the value for success from FSRW methods; errors are returned as negative numbers). The name of the file is converted to uppercase with **.ucstr()** to make the rest of the process a little easier.

The next step is to look for the presence of the ".WAV" extension in the file name; the **instr()** method will return a value of zero or greater if this is the case. If the extension is present, we remove it by converting it to zeroes with **bytefill()**. The position in the list for this file is calculated from the current file count; then, the name is

moved into the list. This continues until there are no more WAV files on the SD card, or we've filled the list with **MAX_FILES** names.

Just to be clear, this list is stored as eight-character names without terminating zeroes (unless the name is less than eight characters) — so, we can't use standard string methods to retrieve an entry.

With a list built, we can randomly play the files from it. Remember, though, the WAV object requires a full file name, and as just pointed out, we've truncated those to save storage space. When the program needs a full name, the **.wav_name()** method is called:

```

pub wav_name(n) | ofs

bytefill(@filename, 0, 13)
bytemove(@filename, @WavList + (n << 3), 8)
ofs := strsize(@filename)
bytemove(@filename+ofs, @FileExt, 4)

return @filename

```

We start by clearing the *filename* buffer with **bytefill()**. The name saved during the inventory process is copied



As low as...
\$9.95
 each!
 Two Boards
 Two Layers
 Two Masks
 One Legend

Unmasked boards ship next day!

www.apcircuits.com



The Easiest Way to Design Custom Front Panels & Enclosures

Free
Front Panel
Designer



You design it
to your specifications using
our FREE CAD software,
Front Panel Designer

We machine it
and ship to you a
professionally finished product,
no minimum quantity required

- Cost effective prototypes and production runs with no setup charges
- Powder-coated and anodized finishes in various colors
- Select from aluminum, acrylic or provide your own material
- Standard lead time in 5 days or express manufacturing in 3 or 1 days

**FRONT PANEL
EXPRESS**

FrontPanelExpress.com

from the list into the *filename* buffer using **bytemove()**. Since all names use exactly eight bytes, calculating the position in the list is very easy. Though the maximum length is eight characters, we very well could have a file with a shorter name (it was padded with zeros during the inventory step). By using **strsize()**, we can determine the length of the name once in the *filename* buffer and append the .WAV extension to it, again using **bytemove()**. Finally, this method returns a pointer to *filename* which is passed to the WAV object for playback.

Spin has another built-in function for working with strings: **strcomp()**. This compares two strings and will return **true** if they're equal. I often use this to return the position index of a string within a known list. Let's say, for example, that we have an RFID application that knows about five tags that are built into our program:

```
dat { tags data }

Tags
Tag0      byte      "0415148F26", 0
Tag1      byte      "0415148E0C", 0
Tag2      byte      "041514AEA3", 0
Tag3      byte      "041514A076", 0
Tag4      byte      "04129C1B43", 0
```

This table matches the RFID string returned from the reader (minus the framing bytes) which is stored in *tagbuf*. Here's a method that uses **strcomp()** to find an RFID tag string in a list like this:

```
pub tag_index(p_tags, tmax) | idx

idx := 0

repeat while (idx <= tmax)
  if (strcomp(@tagbuf, p_tags))
    return idx
  else
    ++idx
    p_tags += 11

return -1
```

Note that this version is specific to RFID tags; hence, the strings are all 10 bytes; this explains the += 11 in the **else** clause (10 bytes plus terminator). One of the advantages to fixed-length strings is that they can be replaced while the program is running, as might be the case with an RFID tag database.

Moving on, can we generalize this code for variable length strings? Of course! Before we do, let's consider these strings:

Jon McPhalen

JON MCPHALEN

Are they equal? Our human side says, "Yes," while our tech expert side wags a finger and says, "No!" — and the tech side is correct. Okay, then, is there a way to compare strings regardless of case? Yes, but we have to do it manually. No strings library is complete without methods to change the case of characters and strings:

```
pub ucstr(p_str)

repeat strsize(p_str)
  byte[p_str] := upper(byte[p_str])
  p_str++

pub upper(c)

if ((c => "a") and (c <= "z"))
  c -= 32

return c

pub lcstr(p_str)

repeat strsize(p_str)
  byte[p_str] := lower(byte[p_str])
  p_str++

pub lower(c)

if ((c => "A") and (c <= "Z"))
  c += 32

return c
```

If the program will tolerate modifications to the strings we want to compare, we can run them both through **ucstr()** or **lcstr()** and then use **strcomp()**. If modifications will cause a problem, then we do it character by character:

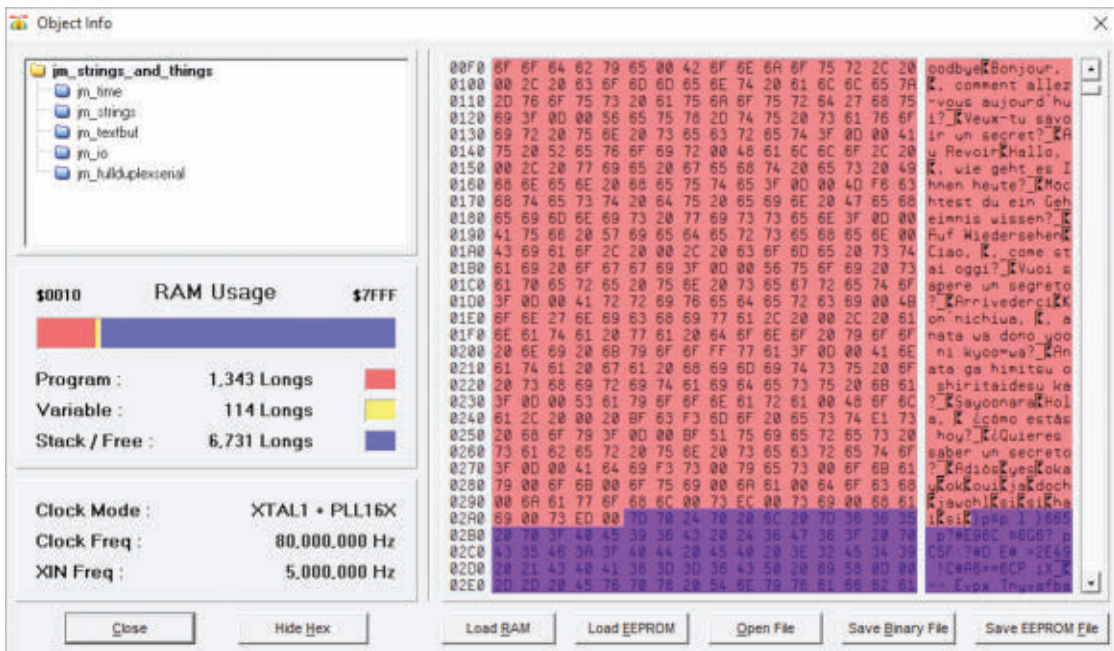
```
pub nc_strcomp(p_str1, p_str2)

if (strsize(p_str1) <> strsize(p_str2))
  return false

repeat strsize(p_str1)
  if (upper(byte[p_str1++]) <> upper(byte[p_str2++]))
    return false

return true
```


■ **FIGURE 1. Encoded strings.**



Jon "JonnyMac"
McPhalen
**jon@jonmcphalen.
com**

Parallax, Inc.
www.parallax.com

You might wonder why I bother checking the length. For long strings, this can save time. For short strings, it does add some overhead, but I think it's worth doing. Putting all this together, we can create a list search that isn't case sensitive:

```
pub nc_str_index(p_str, p_list, lcount) | idx

    idx := 0

    repeat lcount
        if (nc_strcmp(p_str, p_list))
            return idx
        else
            ++idx
            p_list += strsize(p_list) + 1

    return -1
```

This simple method can be used for converting a piece of text to a numeric value that can be used to direct the flow of a program.

String concatenation is frequently asked about in the forums. We've seen it, but let's just go through it again. First, the result array must be big enough to hold the characters of the concatenated strings plus the zero terminator. By combining **strsize()** and **bytemove()**, we can easily concatenate strings:

```

bytemove(@obuf, @ibuf1, {
} strlen(@ibuf1)+1)
bytemove(@obuf[strlen(@obuf)], {
}@ibuf2, strlen(@ibuf2)+1)

```

Okay, it's easy, but a tad tedious for my liking. I have a fairly large strings library that is based on C libraries, as well as code written by other Propeller programmers. That said, most of the time I need simple buffer handling, so I created a separate object. I'm not worried about the overlap in functionality. Hard as we try, I think it's impossible not to have some overlap in related libraries.

The text buffer object does not, in fact, contain a byte array to hold the text. What we do in our application is define one or more buffers, then use the `.set()` method of the text buffer object to link it. Using the array defined above, we could do this:

```
tbuf.use(@buffer, 15)
tbuf.set(@ibuf1)
tbuf.puts(@ibuf2)
term.str(tbuf.address)
```

The first line assigns *buffer* to the object, with a maximum length of 15 characters. The second line sets *buffer* to the contents of *ibuf1*. The third line appends *ibuf2* to *buffer*. The last line sends the newly concatenated string to the terminal window. That's quite a bit easier on the eyes — and the brain. Don't you agree?

Before we wrap up, let me show you how I accept a string from a serial/terminal device:

```
pub get_string(p_str, cmax) | len, c

len := 0

repeat
  c := term.rx
```

```

case c
  8 :
    if (len > 0)
      len -= 1

  13 :
    quit

  32..126 :

```

```

if (len < cmax)
  byte[p_str++] := c
  len += 1

byte[p_str] := 0

```

As you can see, this is quite easy. We need to pass a pointer to the buffer to hold the string, as well as the maximum number of characters that can be entered. A

repeat loop processes the input as long as there is room in the buffer. I do a lot of terminal based work, so I included the ability to handle the backspace key to fix a mistake.

For you fans of the hit TV show, *Mr. Robot*, the penultimate episode from Season 2 was pretty interesting because there was a major shout-out to my friend, Ryan (1o57) Clarke, and the crypto work he did with the Propeller-powered DEFCON 22 badge. There are many strings stored in the badge code that serve as clues in his crypto puzzle. Knowing that DEFCON attendees would start by writing a RAM based program to dump the EEPROM contents to a terminal, Ryan encoded the strings using a few techniques. Well, there was, in fact, one string that wasn't encoded that read:

**DID YOU REALLY THINK THAT
IT WOULD BE SO EASY?**

Have a look at **Figure 1**; this is the EEPROM map from the strings demo program. The highlighted section is a couple encoded strings; these can be decoded with methods in the text buffer object. As a tribute to Ryan and the show, one of the methods is called **.mr_robot()** which runs the ROT-13 algorithm. If you answer a question in the demo program, the strings will be decoded and revealed.

The crypto methods in the text buffer object are not secure, and meant mostly for fun. Still, you can encode and decode string data with them that may dissuade less-than-inspired hackers from looking for information in your EEPROM.

www.allelectronics.com Order Toll Free 1-800-826-5432

24 X 1 LCD - REDUCED PRICE



Wintek # WDC2401P-1GNNA.
Module size: 4.25" x 0.79" x 0.3".
Display size: 3.5" x 0.4". Built-in driver.
14 pin male header, pins on 0.05" centers.
Includes hook-up diagram.

Reduced 50% from \$1.00 each!

CAT# LCD-111

2 for \$1.00

50 for 45¢ each
162 for 40¢ each
810 for 35¢ each

12 HOOK & LOOP STRIPS W/ ADHESIVE BACKING, 4"

4" x 0.90" hook and loop strips with strong adhesive backing. For indoor/outdoor use. Water resistant. Includes both hook and loop. Package of 12.

CAT# HL-412

\$2.50
12-pack

10 for \$2.25
per 12-pack



2-CONDUCTOR AC POWER CORD, SVT ROUND

6' black
18AWG/2 SVT
power cord.
0.23" nominal o.d. cable.
Molded polarized 2-conductor plug to blunt end. UL, CSA.

CAT# LCAC-326

\$1.60
each

100 for \$1.40 each • 500 for \$1.25 each



2" BOSTON ACOUSTICS

2" cone with soft rolled edge. 4 Ohm impedance. Shielded magnet.

CAT# SK-304

\$2.00
each

10 for
\$1.75 each



1 GB MICRO SD CARD

SanDisk Micro SD

CAT# MSD-1

\$2.00
each



10 for \$1.85 each • 100 for \$1.70 each

BASEBALL CARD FRAME

Unique, free standing acrylic photo frame. Two crystal clear acrylic blocks held together by neodymium magnets. Great for displaying baseball cards, business cards or photos.

Trapezoid shaped, it can stand on any of its four edges. 5.95" x 5.93" x 0.95".

CAT# PF-2



\$4.95
each

10 for \$4.50 each

18 AWG STRANDED WIRE

35' roll of 18 gauge stranded, black, multi-purpose wire. Suitable for home automotive and electronic project use. UL, CSA.

CAT# 18BK-35

\$4.30
per roll



INFRARED SENSOR

RE 200B is a passive infrared sensor designed to pick up heat radiation in the infrared range. Used in alarm systems and automatic switches that respond to human body motion. TO5 metal case w/ 3 leads.

CAT# PIR-200B

\$1.00
each

10 for
90¢ each



Item

Propeller Activity Board

Source

Parallax #32910

Happy Holidays, Friends!

As this is my last [scheduled] column for 2016, please allow me to thank my friends at *Nuts & Volts* (especially Robin, who is very patient with me), my wonderful friends at Parallax (in particular, CEO Ken Gracey), and all of you who read my column and provide such nice feedback. I sincerely appreciate it.

I know this column is called "The Spin Zone," but I think in coming months we will start exploring C as well. Plus, PropBASIC is back, so we should touch on it again. Heck, we might even experiment with Forth. A nice bloke from Australia named Peter has created a cool implementation called Tachyon that is very interesting. This is the wonderful thing about the Propeller and why it's my first choice: It's powerful, yet easy to program in a variety of languages.

This is the time of year where we tend to give gifts. If you're looking to get a youngster involved in tech, the new Scribbler 3 from Parallax is an excellent way to go. It's powered by the Propeller, and can be programmed by newcomers using Blockly, which is a visual editor. Of course, it can also be programmed in Spin (the base code already is), C, or any other language supported by the Propeller.

Earlier, I mentioned my friend and acting manager, McKenzie. We met because her young son, actor Zach Rice saw my DEFCON 22 badge with WS2812s and became very excited, telling me in very certain terms that he wanted to learn how to build things like this. I'm

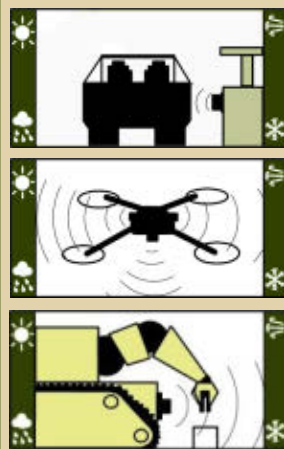
mentoring Zach using the Scribbler and he's having a great time. Pretty soon, I won't be the only actor in Los Angeles who's an expert in Parallax products!

From my family to yours, Happy Holidays, and may God bless you all. Until we meet again in 2017, keep Spinning and winning with the Propeller! **NV**

CLASSIFIEDS

ROBOTICS

Need sensors?



www.maxbotix.com

LIGHTING



HARDWARE WANTED

DEC EQUIPMENT WANTED!!!

Digital Equipment Corp.
and compatibles.
Buy - Sell - Trade

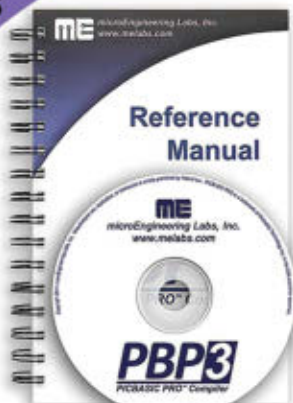
CALL KEYWAYS 937-847-2300
or email buyer@keyways.com



www.melabs.com

Development Tools and Learning Systems
PICmicro® microcontrollers

Buy Now!



- PICBASIC PRO™ Compiler 3.0
- M.E. Labs Trainer
- U2 Programmer
- Prototyping Boards
- Experimenter Boards



Special Price of **\$149.99**
Use Coupon Code **NV1116**

Valid until December 31 2016
(Retail price \$189.99)

Academic and bulk pricing available

PICBASIC and PICBASIC PRO are trademarks of Microchip Technology Inc. in the USA and other countries. PIC is a registered trademark of Microchip Technology Inc. in the USA and other countries.

Nordic's New nRF52832 Gets You On Track with BLE Apps

The number of microcontrollers being repackaged as SoC (system-on-a-chip) packages is becoming common place. You can get on the Wi-Fi wagon with an ESP8266 module, which can either be hosted by an external microcontroller or programmed natively using free compilation tools. Many of the new eight-bit Super PICs are being loaded with operational amplifiers, digital-to-analog converters (DACs), and a plethora of analog subsystems. The PSoC folks have been pushing the system-on-a-chip for quite some time.

This installment of Design Cycle will feature the new Nordic Semiconductor SoC. The Nordic SoC includes all of the analog features found in many of today's microcontrollers, plus a full functional BLE (Bluetooth Low Energy) radio.

Required Development Tools

The newest Nordic BLE device is the nRF52832. You can use the nRF52832 just like any other microcontroller as its core is an ARM Cortex-M4F. The nRF52832 is equipped with 512 kB of Flash program memory and 64 kB of RAM. For BLE work, the nRF52832 calls upon the services of a Nordic SoftDevice.

A SoftDevice is a precompiled block of code that contains a full BLE stack, which can be accessed using a set of API calls. The architecture of the nRF52832 is such that the SoftDevice resides in a protected Flash memory area. The remaining Flash can be used for the application code. The nRF52832 is designed to be loaded with SoftDevice 132, which is available free of charge from Nordic Semiconductor.

To load a SoftDevice, you will need nRFgo Studio. The nRFgo GUI is captured in **Screenshot 1**. As you can see,

the nRFgo Studio allows you to load a SoftDevice as well as an application and bootloader. In our case, SoftDevice 132 is loaded and our user Flash area begins at location 0x1F000.

nRFgo Studio relies on a hardware device to program the nRF52832 device. I prefer to use the "real thing" which is a Segger J-Link Pro. My J-Link is shown in **Photo 1**. The J-Link Pro is top-of-the-line. You can also employ the services of lesser equipped J-Link devices.

The beauty of using a J-Link programming/debugging device is that the Keil MDK-Cortex-M C compiler integrates seamlessly with the J-Link device and its drivers. You will need the Keil Cortex-M MDK to compile and debug your nRF52832 application code.

Once you have coded your nRF52832 BLE application, you will need some way of pulling the radio information in for verification. Monitoring your nRF52832 radio transmissions requires the Nordic Master Control Panel, which is also as close as a download from the Nordic website. The Master Control Panel is fed by hardware. You can use the Nordic NRF51 Dongle to feed the Master Control Panel. The NRF51 Dongle also runs the Nordic Sniffer application which allows you to use Wireshark to gather information on the BLE transmissions. The NRF51

Post comments on this article and find any associated files and/or downloads at www.nutsvolts.com/magazine/article/November2016_DesignCycle_Nordic-nRF52832-BLE.



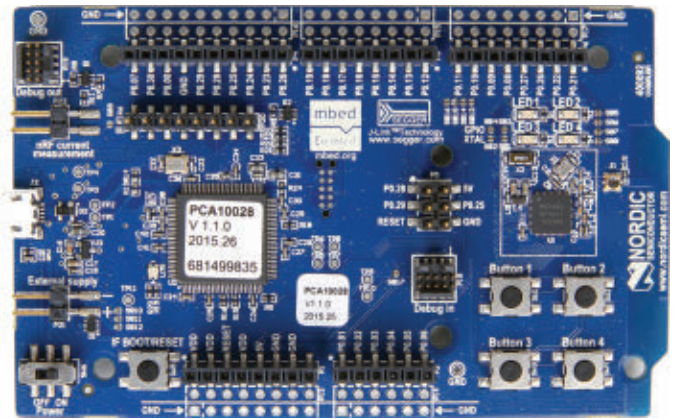
■ **PHOTO 1.** Now that I have it, I don't know what I would do without my J-Link programmer/debugger. This puppy is right at home with the Nordic ARM Cortex-M devices. The J-Link can even be used to program and debug certain Microchip PIC devices.

Dongle comes standard with Segger OB (On Board), which provides

the programming/debugging services offered by standard stand-alone J-Link devices.

An alternate to the NRF51 Dongle is the nRF51-DK. The nRF51-DK is equipped with Segger OB and can be loaded with the Master Control Panel firmware directly from the Master Control Panel PC application. The nRF51-DK you see in **Photo 2** was used to gather some of the information you will see in this discussion. The nRF51-DK can also be programmed to act as a standard BLE device or a BLE Sniffer.

The aforementioned Nordic development hardware and firmware is designed to integrate with the Nordic SDK. The current Nordic SDK is version 12. Everything you will need to assemble an nRF52832 BLE application using the Nordic tools we have

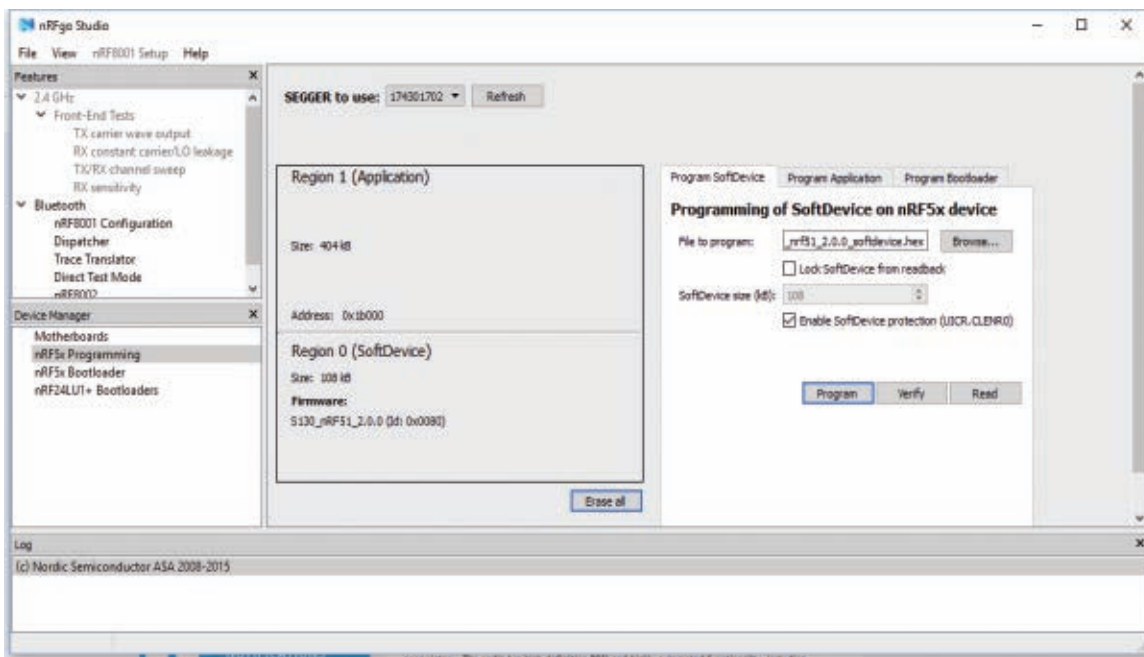


■ **PHOTO 2.** If you plan to work with Nordic BLE devices, this nRF51-DK is a must-have development tool. As of this writing, the new nRF52832-DK doesn't support all of the PC based development tools.

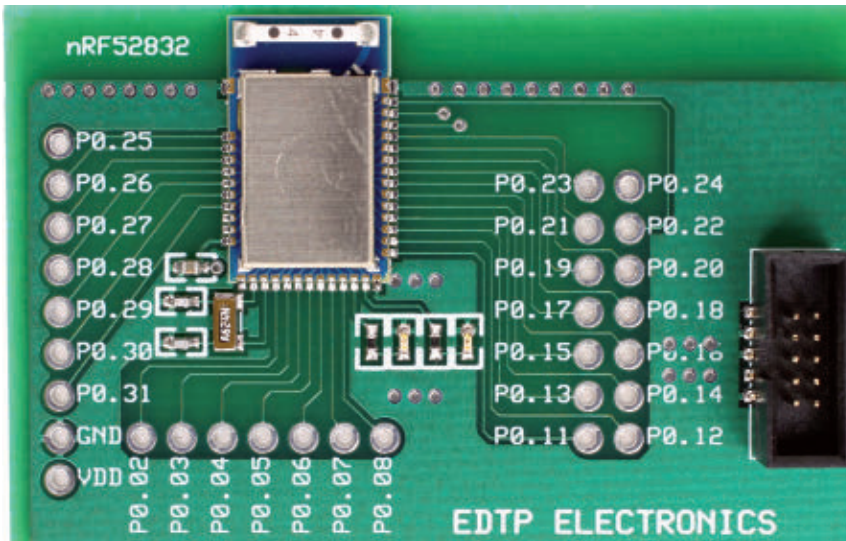
discussed is contained within the new SDK.

Our BLE Hardware

The Nordic nRF52832 was set up to be designed into the fabric of a device. Not all of us have the capability to lay down that tiny nRF52832 QFN package, however. Not to worry. Nordic has an extensive following of third party vendors that offer the raw nRF52832

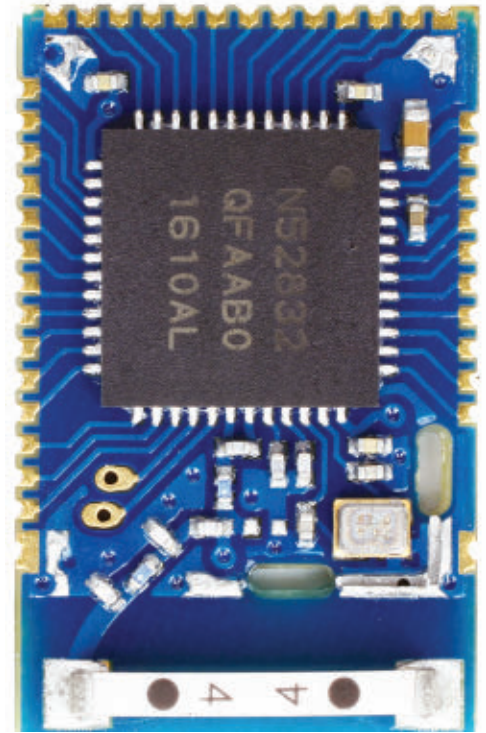


■ **SCREENSHOT 1.** The nRFgo Studio follows Nordic's super easy approach to BLE development. As you work with Nordic devices, you will find that every Nordic development tool is a breeze to use.



■ **PHOTO 3.** The design is realized on a standard double-sided printed circuit board (PCB).

embedded in an easy-to-use module. I have chosen the Raytac module because the Raytac folks were “Johnny on the spot” with their nRF52832 module, which is referenced in the Raytac documentation as the MDBT42Q.

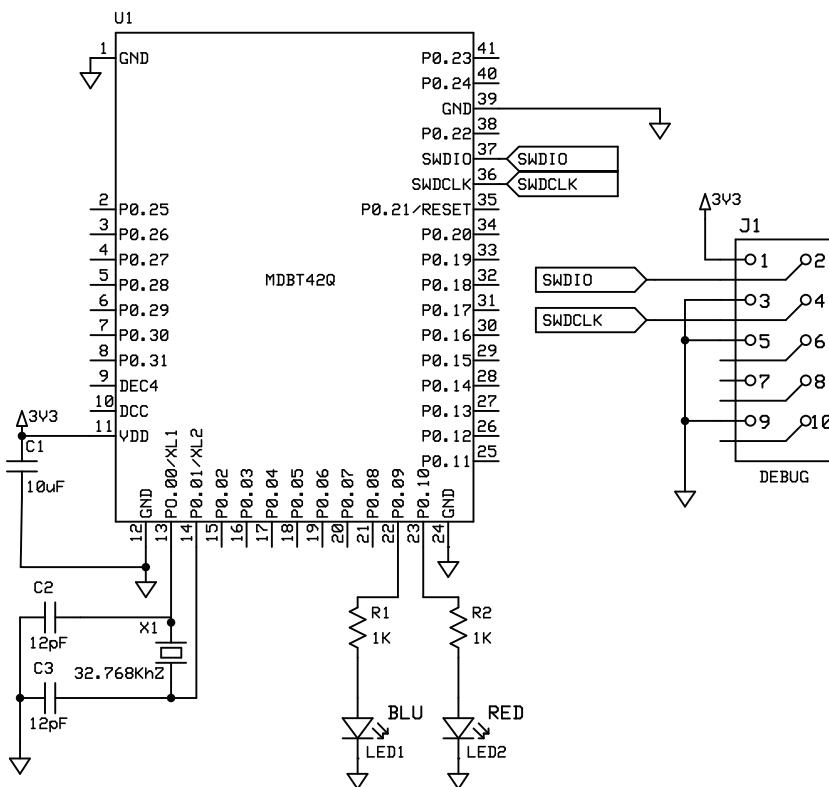


■ **PHOTO 4.** The MDBT42Q is a standard nRF52832 circuit that has been FCC certified and placed within an easy-to-mount package.

I have designed and assembled an MDBT42Q development platform that we will use to bring our BLE application to life. The MDBT42Q design is depicted graphically in **Schematic 1**.

The MDBT42Q development board in its silicon and fiberglass form is under the camera in **Photo 3**. The 32.768 kHz crystal is a must for low power operation. The SoftDevice in particular utilizes the 32.768 kHz crystal. The ARM core of the MDBT42Q runs on the 64 MHz clock. You can see the 32 MHz crystal that drives the MDBT42Q’s 2x PLL in the uncovered module captured in **Photo 4**.

Take another look at **Photo 1** and you will see that the J-Link Pro is sporting an OLIMEX JTAG 20-10 adapter. This adapter allows us to mount a 1.27 mm pitch 10-position debug/programming header on our MDBT42Q development platform.



■ **SCHEMATIC 1.** This schematic is simple because all of the magic resides inside of the MDBT42Q module. All of the free GPIO pins are terminated at a header point on the PCB.

Coding Our Board Support Package

The nRF5-SDK has an excellent built-in board support mechanism. To utilize the SDK's BSP services, all we have to do is create a file called *custom_board.h* which contains the physical GPIO mapping of our LEDs. As you can see in **Schematic 1**, our MDBT42Q development board includes a pair of blue and red LEDs. One of the coolest features of the nRF52832 is that we can place any microcontroller function on any of the MDBT42Q's GPIO pins we desire without restrictions. However, in this case, the pins I selected for the LEDs are reserved out of the box for the NFC feature of the nRF52832. To use the NFC pins as general-purpose GPIO pins, we must place a statement in the compiler preprocessor list.

While we're modifying the preprocessor list, we must also specify to the C compiler to use our special BSP file. The preprocessor modifications are shown in **Screenshot 2**. Here's what the *custom_board.h* file LED entry looks like:

```
#define LEDS_NUMBER      2
#define LED_START        9
#define LED_1            9
#define LED_2            10
#define LED_STOP         10
#define LEDS_LIST { LED_1, LED_2 }
#define BSP_LED_0        LED_1
#define BSP_LED_1        LED_2
#define BSP_LED_0_MASK   (1<<BSP_LED_0)
#define BSP_LED_1_MASK   (1<<BSP_LED_1)
```

```
#define LEDS_MASK (BSP_LED_0_MASK | BSP_LED_1_MASK)
// all LEDs are lit when GPIO is low
#define LEDS_INV_MASK LEDS_MASK
#define BUTTONS_NUMBER 0
```

Another thing you will discover about developing with Nordic tools is that there is a working template for everything. For instance, our *custom_board.h* file is simply an edit of the existing *pca10040.h* BSP file, which is part of the Nordic nRF5 SDK. The *pca10040.h* file maps the LEDs and buttons found on the nRF52-DK development board. Our *custom_board.h* file maps our garage-brewed development board's pair of LEDs to the MDBT42Q's GPIO pins 9 and 10.

Out-of-the-Box BLE

The Nordic nRF5 SDK is chocked full of working examples and operational ready-to-run templates. The *ble_app_template* included in the nRF5 SDK contains enough code to enable the GAP and Link Layer parameters that are required to put the nRF52832 radio to work.

For those of you that are a bit BLE challenged, GAP is short for Generic Access Profile. In a nutshell, GAP is all about the connection. GAP defines whether a device will act as the Central (cell phone, tablet, etc.) or Peripheral (our nRF52832 development platform). GAP also defines the Broadcaster and Observer roles. A Broadcaster conveys data by advertising and does not connect to receive

incoming data. The Observer assumes the scanner role and simply listens. Observers do not transmit.

The Link Layer is built around a state machine that rotates on roles (Master and Slave). Link Layer states include scanning, standby, advertising, initiating, and connected. The Link Layer states pair with the roles they play. For instance, an advertiser is the Slave device, while an initiator is the Master device. Slave devices advertise and Master devices scan. The Link Layer roles relate to the GAP roles in this manner:

```
Link Layer Master = GAP Central
Link Layer Slave = GAP Peripheral
```

The exchange of data is controlled by GATT (Generic

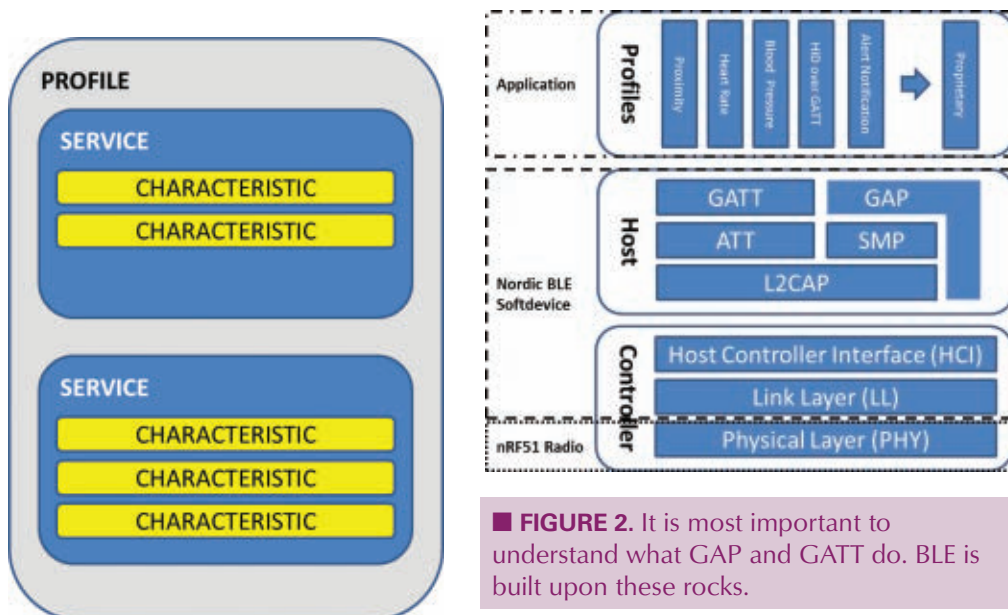
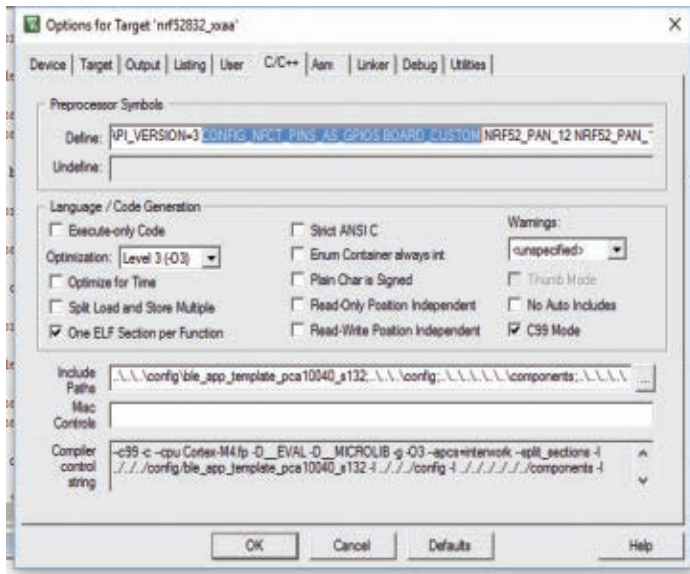


FIGURE 1. Data is carried in the Characteristics. Characteristics are logically grouped in Services. A collection of Services is called a Profile.



■ **SCREENSHOT 2.** We must include the preprocessor symbol `CONFIG_NFCT_PINS_AS_GPIO` to use the MDBT42Q's pins 9 and 10 as standard GPIO. `BOARD_CUSTOM` tells the compiler to use the board configuration entries within our `custom_board.h` file.

Attribute Profile). This data exchange mechanism is based on Services and Characteristics. Simply put, a Service is a

logical collection of Characteristics. A collection of Services is called a Profile. **Figure 1** is a graphical representation of how Profiles, Services, and Characteristics interweave.

GATT is based on the client/server model. Let's expand on our relationship between the Link Layer and GAP:

Link Layer Master = GAP Central = GATT Client
Link Layer Slave = GAP Peripheral = GATT Server

To pull it all together, take a look at the BLE stack drawn up in **Figure 2**. GATT is based on ATT (Attribute Protocol). ATT can be thought of as a lookup table of Services and Characteristics that are uniquely identified using 16-bit IDs.

Let's quickly examine the **Figure 2** BLE stack bottom to top. The Physical Layer is the nRF52832 radio. The PHY is responsible for converting analog data to digital data and passing it along to the Link Layer. The PHY is also capable of processing digital data it receives from the Link Layer and transmitting it over the air.

The Link Layer is there to oversee scanning, advertising, and creating/maintaining connections. L2CAP (Logical Link Control and Adaptation Protocol) performs the transportation of data for higher layer protocols, including multiplexing multiple applications over a single link. SMP (Security Manager Protocol) is obviously used for security of the BLE link and data.

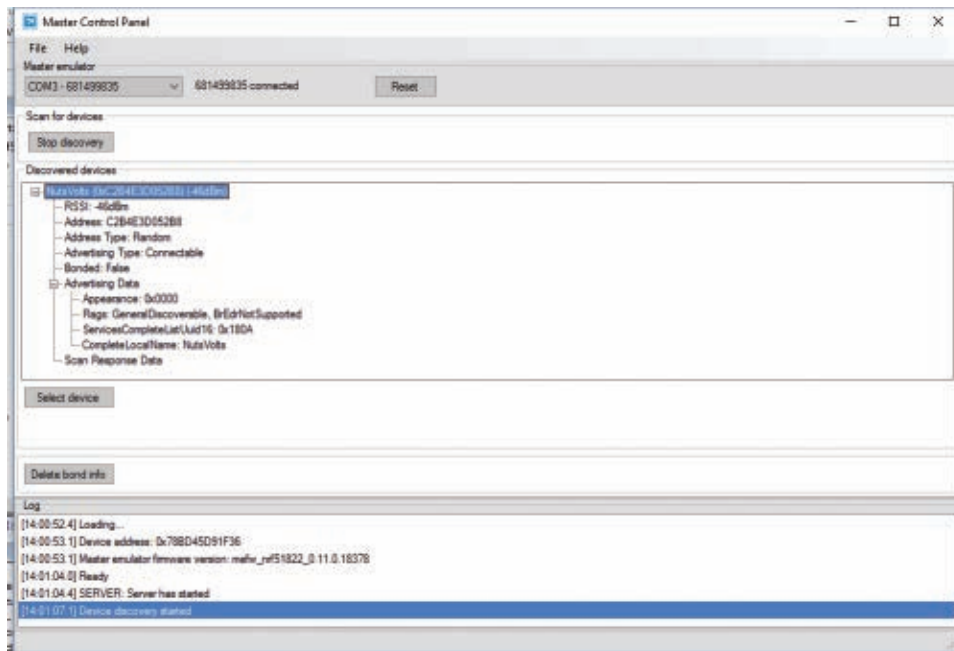
I'm not going to list the template source code here as you can get the source by downloading the version 12 nRF5 SDK. Once you have a copy of the `ble_app_template` project code, you will see that the GAP and associated connection parameters have already been defined. The same goes for the advertising functions.

There are a couple of changes that we will make just to show that the code is working as designed. Let's change the device name and eliminate the advertising timeout:

```
#define DEVICE_NAME
"NutsVolts"

#define APP_ADV_TIMEOUT_IN_SECONDS 0
```

Earlier, we used nRFgo Studio to load SoftDevice 132. So, all we need to do at this point is compile the template code with our changes.



■ **SCREENSHOT 3.** Master Control Panel has picked up our nRF52832's advertisement. We know this is ours as the device name (NutsVolts) reflects the change we made in the original template source code.

The code should compile with no errors. The resultant hex file is then downloaded to our MDBT42Q module. An indication that our development board is working as designed is the constant on state of the red LED while the blue LED is blinking. This indicates that our designated custom BSP is being used and that our MDBT42Q is advertising. **Screenshot 3** is our absolute verification.

The advertisement that was detected by our nRF51822-DK hardware and passed to the Master Control Panel contains our device name (NutsVolts). The six bytes that follow the device name contain the MDBT42Q's universally unique hardware address.

Screenshot 3 also tells us that we can connect to our peripheral device, but there are no obvious data-bearing services or characteristics being exposed by the advertisement. Thus, there is no user-defined data on the GAP Peripheral/GATT Server (our MDBT42Q board) that can be accessed by a GAP Central/GATT Client (cell phone or tablet).

Let's connect anyway. There are some Services and Characteristics present that are holding data that is relevant to the connection. In **Screenshot 4**, you can see that our device name can be found in slot 3 of the GATT/ATT based attribute table. We also see that the device name is associated with a UUID of 0x2A00. This is by design. Every device name will be identified with a UUID of 0x2A00.

The same holds true for each UUID you see in **Screenshot 4**. These are assigned UUID type definitions that are laid out in the BLE specification. Note that each Characteristic is preceded by a Characteristic Declaration, which provides information about the Characteristic value. This is standard BLE operating procedure when it comes to Services and Characteristics.

We're Not Done Just Yet

We have proven that the Nordic nRF5 SDK paves the way for easily creating and implementing BLE applications using the new nRF52832. The next step is to add a Service to our BLE application. Once our Service is in place, we can populate it with a Characteristic.

So, get your hands on the tools that were referenced in our

SOURCES

Nordic Semiconductor

nRF52832

Master Control Panel

nRFgo Studio

nRF51-DK

nRF52-DK

www.nordicsemi.com

Segger

J-Link Pro

www.segger.com

Raytac

MDBT42Q

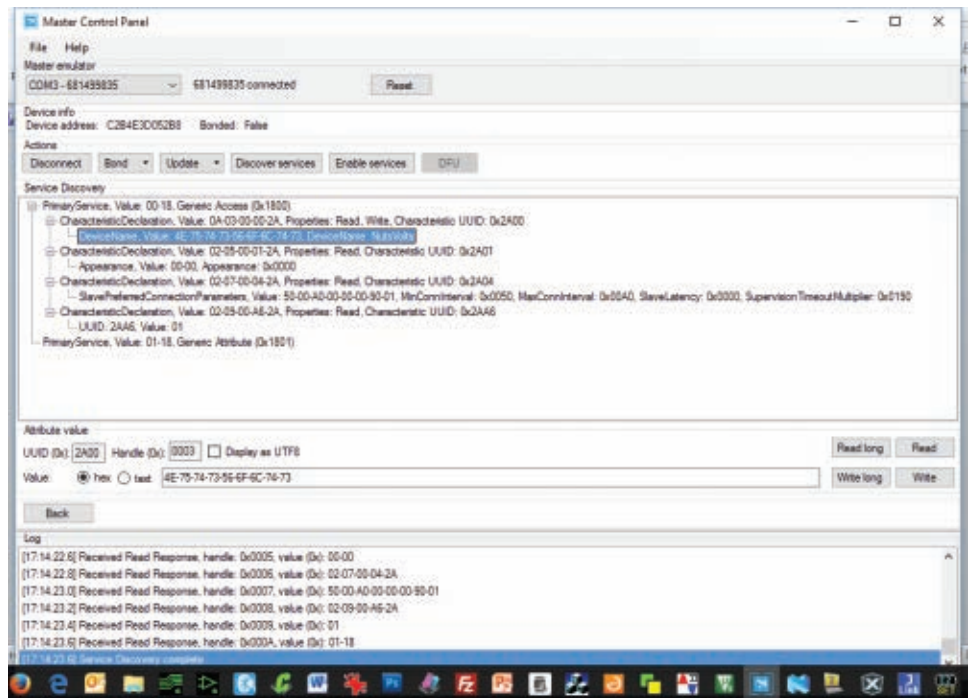
www.raytac.com

Keil

MDK-ARM-Cortex-M C Compiler

www.keil.com

discussion and build up your version of the Raytac dev board. We will continue this thread in next month's Design Cycle. **NV**



■ **SCREENSHOT 4.** The scanner (GAP Central, GATT Client, Link Layer Master) picks up the advertisement, and based on the advertisement information makes a decision whether or not to connect to the advertising device (GAP Peripheral, GATT Server, Link Layer Slave). The device name is number 3 in the GATT table (Handle). The universal attribute type given to the device name is denoted by 0x2A00. The value of the device name characteristic is "NutsVolts."

HIGH EFFICIENCY/HIGH POWER PROGRAMMABLE DC SUPPLIES

Saelig Company, Inc., introduces the PSU-Series of single output programmable switching DC power supplies that can provide up to 200A current and cover a power range up to 1,520W. These high efficiency / high power density supplies include five models from 6V to 60V rated voltages. The PSU-series can be connected in series for increased power capacity – up to two units in series or up to four in parallel to handle a broad coverage of applications.

- PSU 12.5-120 provides 12.5V at up to 120A
- PSU 20-76 provides 20V at up to 76A
- PSU 40-38 provides 40V at up to 38A
- PSU 6-200 provides 6V at up to 200A

The PSU-Series of 19" rackmount (1U) single-channel power supplies from GW Instek provide constant voltage / constant current selection which is a very useful safety feature for protecting a Device Under Test (DUT). The power supplies normally operate in constant voltage (CV) mode when turned on, but, if connected to a capacitive load, this could cause a high inrush current or current-intensive load at the power output stage. Running in constant current mode limits current spikes, protecting the DUT from inrush current damage. The over-voltage (OVP) and over-current (OCP) protection levels can be selected



from 10% to 110%, with the default level set at 110% of the power supply's rated voltage / current.

The adjustable slew rate of the PSU-Series allows users to set either output voltage or output current with a specific rise time for low to high level transition, and a specific fall time for high to low level transition. This facilitates the characterization of a DUT during voltage or current level changes with controllable slew rates.

The Output On/Off delay feature enables the setting of a specific time delay for Output On after the power supply output is turned on, and a specific time delay for Output Off. When multiple PSU units are used, the on/off delay time of each unit can be set respectively at preset time points. This multiple-output control can be done through the analog control terminal at the rear panel or through PC programming with standard commands.

The PSU-Series provides USB Host, USB Device, LAN, RS-232 with RS-485, and isolated analog control interfaces as standard. Pricing begins around \$2,200. A LABView driver is also available for rear-panel external control of power on/off and external monitoring of the power output voltage and current.

For more information, contact:
Saelig Company, Inc.
www.saelig.com

WAILER WAH PEDAL

Electro-Harmonix introduces the Wailer Wah which features the circuitry, sound, and tone of the award-winning Crying Tone in a traditional rack and pinion style pedal, and at an extremely affordable price point.

The Wailer Wah delivers expressive wah and cocked wah sounds and – at just over 1.5 pounds – a substantial

weight savings over many popular wah wah pedals.

Electro-Harmonix President, Mike Matthews, stated: "Our goal was to build a wah pedal that sounds great, deliver it at an astounding low price, and make it provide good weight savings for the player who has to schlep their own gear around. I am very excited about the EHX Wailer Wah."

The Wailer Wah comes equipped with a 9V battery or can be powered by an optional nine volt AC adapter. The pedal carries a street price of \$62.75.



For more information, contact:
Electro-Harmonix
www.ehx.com

I almost completed a Columbia University Electronics course when I was discharged from the Army in 1968. By 1970, I had completed my RADAR section and was going into advanced studies (engineering) when I had to go to work full time and simply too many distractions stopped me in my tracks (got married, had kids, new home, car payments — the usual young people struggles).

However, because of that electronics training, I landed a job in a steel foundry as a lab technician. I did so well with the math, the company put me on through college in the specialty study of metallurgy. "Specialty" means not a degree course, but designed to train me in specific control of several types of nuclear certified steel castings.

I was given the great opportunity of study in universities in Seattle (International Harvester sponsored some excellent courses up there) and in Milwaukee, WI and Chicago in university studies, and practical work in some of the most modern steel foundries in the nation. I became the melt supervisor and chief metallurgist for my foundry in California.

Eventually, California's ever-tightening environmental laws ruined the steel industry in that state and I lost my job when the company closed, and moved my family to another state, and never went back into the steel foundry industry.

Fortunately, my electronics training has given me a lifetime of fun and better than average understanding of the world around me. It always amazed me how much electronics math and algebra and the pure physics of electrical science applies to so many other disciplines. Engineering, fluid mechanics, metallurgy, construction, and even social sciences and forestry all use similar math.

Although my electronics training was incomplete, the basics and mid levels of the science has given me a foot-hold and even a "boot-up" on just about any endeavor I ever

wanted to try.

Anyway, I cannot even imagine a life without a good understanding of electronics. It has been one heck of a fun ride.

Altoidian

My father! He inspired me, but in a kind of roundabout way. My father was an electronics engineer in an era where electronics evolved from tubes through to integrated circuits. He specialized in RF design. I did very poorly in school growing up and didn't even graduate high school, but always, deep down, had an interest in electronics because of my father.

We had a lab in the house as long as I could remember, and I tinkered in there without a great deal of understanding of electronics since I was 10 years old. I'd ask my father questions, but he could never bring the answers down to my level. Because I had no high school diploma, I could not enter university and went into retail for 10 years. I even built my own lab in my house after saving for years and kept the electronics hobby going, teaching myself the basics of analog and digital design and eventually getting into embedded software development.

That deep connection with electronics never went away and with a desire to understand what my father could teach me, I decided to try school once more. At 28, I entered high school again and spent two years earning my diploma, graduating with honors, and earning a scholarship to university in electronics engineering.

After my first year, I applied for and received a position at a technology firm for the summer. At the end of the summer, they offered me a long-term position, part time, while I finished university. I accepted the position and worked half time while at university full time. The desire to be like my father and share our love of electronics kept me going through this very intense time.

I graduated with honors and received an offer of full time employment at the same technology firm. It was that year my father fell ill and as I started to grow my career in engineering at age 34, my father passed away. In the months before that, I spent hours in the hospital with him talking about technical issues, electronics, software, design, and engineering in general. I brought him a computer and we developed software models together (nerdy, I know). I could finally talk his language and understand most things he taught.

On his last day — and for the first time in our relationship — he told me how proud he was of me. He was a genius and an inspiration, and to this day, I try to follow his lead and try to inspire others to join this incredible discipline. I spend time teaching electronics and engineering to grade schoolers, hoping to do better at bringing this field to young folks.

After 20 years, I'm still with the same company and incredibly grateful for the opportunities they've provided. I'm looking forward to the next 20, thanks to my Dad.

Nohj

The ARRL *Handbook for Radio Communications* — better known as the *Radio Amateur's Handbook*. The current edition is a bargain for up-to-date stuff. Buy older editions at hamfests or used book stores if you still build tube gear.

Chip Veres

Would you like to receive *Nuts & Volts'* weekly content newsletter? You have three ways to sign up:

- Visit us on Facebook and click on "Join My List."
- Using your cell phone, send "NVNEWSLETTER" as a text message to 22828.
- Visit the *Nuts & Volts* FAQs to sign up at nutsvolts.com/faqs

>>> QUESTIONS

Desktop Pi?

After the recent “forced” update to Windows 10, I am so frustrated with big computers and big software companies. I’m thinking of downsizing to the smallest, simplest computer I can manage.

Has anyone successfully replaced a desktop PC with a Raspberry Pi?

#11161

David Grear
Temple Hills, MD

GFI Tester

I have a GFI type breaker for my garage door that trips occasionally, making it so the door won’t open with the remote. I’d like to know how to test the GFI to see if it has gotten weak or failed in some manner.

What’s the best method to test it?

#11162

Henry Doran
Phoenix, AZ

Auto-Headlights

I sometimes forget to turn the headlights off on my car, so I wake up to a dead battery and a late start to work. I would like to build a DIY auto headlight on/off switch. Anyone have a schematic or design?

#11163

Daniel Lemann
Milwaukee, WI

Video Signal Synchronizer

I come from a TV engineering background, so I know I am asking for a lot. This is a very preliminary step in a design process, so I know it may go nowhere.

What I want to do is simultaneously view a small object/area (~ 1/4”) from two angles that are 90 degrees apart. Imagine a straight pin stuck into a board. I want to have magnified views of that point from the front and one side at the same time. These views will be at an angle above the board — perhaps 20 to 30 degrees — but they will be 90 degrees

apart as viewed from above.

One way that I can think of doing this would be to use two of those small video cameras that are widely available for under \$50. I do not need color images; black and white would be just fine. Two monitors would be too much, so I need a way of combining the two unsynchronized video signals to be displayed on a single screen, side by side.

What I am looking for is a combination of a frame synchronizer and a special effects generator that can do a split screen in one or a set of chips. I would think that anything current would be digital in nature, but an analog circuit would also be acceptable. Does anyone know of such a chip or set of chips?

Relatively inexpensive video surveillance systems can display multiple images on a single monitor, so there must be something out there. Of course, price is an important consideration. I would like to keep the entire project to under \$200.

The only other alternative I can think of is to use mirrors to combine the two images optically and use a single camera, but that also has a lot of complications and expenses.

#11164

Edward Alciatore
Beaumont, TX

Tube Tech

Is there any *technical* difference between tube amp distortion and solid-state amp distortion? I have heard tube amps described as “warm” sounding but I can’t find any info as to why. Isn’t “clipping” just “clipping” no matter the device that is performing that function?

#11165

Alison English
Tampa, FL

Power Supply Woes

The power supply in my 1994 Packard Bell Legend 18CD computer has died. I really need

to get on this machine to recover my data. Any ideas where I might find a replacement supply or troubleshooting suggestions?

#11166

Mattias Edvardsen
Warren, OH

>>> ANSWERS

[#8162 - August 2016]

Temperamental Trailer

The lights on my boat trailer work erratically. Sometimes they work fine (turn signals, brakes, and running lights); other times, when I press the brake pedal, only the right turn light comes on and all the running lights go out! Short of tearing it all out and re-wiring from scratch, any tips on how to locate the fault and fix it?

#1 No doubt about it, trailer lights are a pain. There are a multitude of possible trouble points.

First of all, if you are launching your boat correctly, then the back end and sometimes the whole thing gets dunked under water. The channels of the trailer frame get filled with water and never really have a chance to fully dry out. Since the channels are steel, they begin to rust. Rust turns clean shiny ground connections into high resistance points, and strange things begin to happen. The drain holes allow little critters access to the inside of the channels where they set up housekeeping which includes munching on your wires.

The cable feeding the lights from your vehicle gets a lot of flexing and can fail in a whole bunch of ways — especially that flat four-conductor stuff. The lampholders on the trailer are also prone to failure from the periodic dunkings — even the ones claiming to be sealed.

And do not forget the connectors themselves. The vehicle mounted connector is vulnerable to all kinds of weather conditions 365 days a year, which means it is susceptible to

Send all questions and answers by email to forum@nutsvolts.com
or via the online form at www.nutsvolts.com/tech-forum

oxidation and corrosion.

I have to warn you right now that you have to be careful when working on the vehicle connector — especially if it is the large, round seven-contact type. One of the contacts has full vehicle battery voltage on it continuously, protected by a 30 or 40 amp fuse. If you accidentally short this contact to ground, additional damage can result.

So, disconnect the battery first, or at least pull the fuse. **Diagnosis:** First thing you have to do is isolate whether the problems are vehicle or trailer related. Could be both. The auto parts store will sell you a nifty little tester with several LEDs that you plug into the vehicle, and will indicate whether or not a voltage appears on a specific contact when the corresponding circuit is energized.

These can be useful, but have their limits. An LED is a low current device compared to the current drawn by your lights. If you have a less-than-perfect connection somewhere in the system, the LED will merrily light up indicating everything is wonderful. Substitute a real world load of a few amps, and the resistance of the poor connection will drop the available voltage dramatically. Same goes for using your multimeter: The current drawn by the meter is way too low for high resistance connections to have much of an effect. What you can do, however, is connect the trailer to the vehicle, then start checking various points with the meter since the trailer is the real world load condition for the circuit.

Establish a good connection to the vehicle battery negative terminal (assuming a negative ground system) for the minus side of the meter. Then begin checking the various points on the trailer while the lights are energized.

You can also check for crummy ground connections by touching

the meter probe to the shell of each lamp. You should read no more than a few millivolts from the shell to ground. Anything greater than a volt is definitely suspect.

What I ended up doing is building a box that connects between the vehicle and the trailer, with a lamp (incandescent, for the reason cited above) for each circuit so I can see at a glance whether the circuits are getting power from the vehicle or not. The box also allows me to connect a stand-alone battery to the trailer and power each circuit individually for testing purposes. (See diagram at www.nutsvolts.com/tech-forum/question/temperamental-trailer.)

Jerry McCarty
Jackson, MI

#2 The answer is simple: Fix the grounds. The reason that one light lights is that the one light that lights has the lowest resistance connection to the power source (brake light circuit in the car) and without a secure ground connection, the circuit through the lamp is completed through the filaments of all the other lamps back to ground.

Especially with boat trailers, you cannot rely on the metal frame of the trailer for a secure ground. Corrosion between the steel parts insulates them from each other. Similarly, you cannot rely on the trailer ball providing a good contact to vehicle ground. Therefore, you have to have a good source of ground in the vehicle to the trailer light connector, and the connector on the trailer end should be firmly bonded to the trailer frame.

For best results, there should be a ground wire that connects to that same point on the trailer frame going to each light on the trailer.

George Andersen
New Port Richey, FL

[#8163 - August 2016]
Is It Dumb To Have A Smart TV?

Is it possible (or even necessary) to try to protect my "Smart" TV from hackers? If so, what are the issues and where is the best place to start?

#1 YES. Some Samsung and maybe even LG, TVs have voice recognition and cameras. Samsung even states in the user's manual that their TVs may capture personal room conversations of a private nature and transmit them to a third party.

Samsung's suggestion to prevent this is to turn off the audio capture feature (which would then prevent you from giving oral commands to your TV) and/or unplugging the Internet. A piece of tape over the camera would ensure your privacy here. You can read more here: www.theblaze.com/stories/2015/02/09/owners-of-samsung-smart-tvs-should-be-aware-of-this-very-scary-privacy-policy.

Ray
Vancouver, Canada

#2 A television set is a data receiving device (or a "Write-Only-Memory" as the joke that was distributed in the past) such that it has no useful data stored for a hacker to use. If you are unsure about your home network security, then the first item to protect is the local network router, and then the Wi-Fi access point that your "Smart TV" connects to.

The best solution is simply disconnect the TV from the network, and use only local media sources (DVD, VCR, etc.), or broadcast signals (antenna).

Raymond J. Ramirez
Bayamon, PR

[#8164 - August 2016]
How Long Is Nine Volts?

Is there a simple way to determine how long a given circuit will run on a standard 9V battery?

#1 The load current you are drawing from your circuit and the time it will take to drain the battery is determined by the capacity of the nine volt battery you're using. How 'long' it will last as you expressed, is the circuit current your device circuit consumes. The average nine volt battery capacity is given in milliamp hour rating (mAh). For common alkaline batteries, it's about 500 mAh. Very simply, 500 milliamps for one hour. This can be expressed differently but equally as half of one ampere for one hour. It's also 25 mAh for 20 hours, or 10 mAh for 50 hours. We simply divide 500 by your circuit's current load in milliamps; each milli-unit, by the way, is 1/1000 of an amp, to obtain the device load duration in actual time.

Unit measurement is important to understanding relationships in science and technology. You can find this load current draw with your device by using a digital multimeter (DMM) and power source like the actual battery and some connectors. I like to use colored alligator jumpers: black to all marked negatives and then red to all positives. You don't have to color code this way, but it helps to remember polarity. Set the meter to the high milliamp setting — most often 200 on low cost meters — just as long as it's well above your expected load. The current setting can be fragile for all meters since this test measurement might involve occasional large current, as in a SHORT or closed connective path which is 'shunted' through the meter's circuitry to express a current measurement. Thankfully, it's fused. When I finish current settings by habit, I always reset my DVM or VOM to 20 volts or higher so I won't need to replace the fuses protecting the meter if I accidentally measure too much current on too low of a setting.

If you have a tech friend helping, you could also use their adjustable power supply. Set the battery voltage, hook up your device (turn the device

on!), and simply read the output current being used. Pretty much the easy way. Like you, I too like to know what current load various battery appliances use, not only to predict run time, but also how big to "size-up" the battery if I wish. I use this setup a lot.

I assumed for ease that the load your device is using is steady and predictable, but maybe your load varies considerably. Time is constant, but the load fluxuates or turns on and off irregularly. You might proceed with some gnarly math, but I'm more inclined to use a KILL-A-WATT type of household wattmeter. You have one, don't you? Then, plug in your power supply source set at nine volts (but not from the battery), then read the watt-hour number over the time span you care to measure. Subtract the efficiency of the type of power supply you're using* from the total. I confess this is new for me. The Kill-A-Watt units are in 1/100 divisional units, i.e., amps, volts, except watt-hours is in tenths. Fair accuracy. I also have a WATTS UP like clone device that comprehensively continuously measures DC power that is connected between a power source and its load. Drone operators like these to monitor battery charging. Solar energy installations benefit from the detailed information they provide, as well.

After experimenting with the AC wattmeter, the small switching transformers show very small wattage unloaded. The estimated 10% loss isn't worth bothering to mention. Ferro-resonant laminated transformers (wall warts) on the other hand are hogs! An 800 mA 12 VDC unit drew 3.1 watts completely unloaded.

*The large and heavy laminate transformer wall warts are about 60% efficient, and the newer lightweight and smaller high frequency switching transformers are up to 90% efficient. So, in a very loose way, you merely subtract the remaining percentage of the wattmeter's total reading to fetch

a general watt-hour reading. (Length as you would say!) Subtract the percentage difference from your total reading.

Michael Greenlee
via email

#2 It depends on the device's current draw and its duty cycle, and also on the battery type. For example, a nine volt carbon-zinc (Leclanché) — the "cheaper" batteries — are rated about 400 mAh, according to https://en.wikipedia.org/wiki/Nine-volt_battery. That, theoretically, means it could power a 1 mA load (e.g., 6,000 ohm resistor in series with a white LED dropping 3V) for 400 hours, or a 10 mA load (e.g., that same LED with a 600 ohm resistor) for 40 hours. These batteries do not have the same capacity for high currents, so they might provide 100 mA for only an hour or two.

The next step up in price are manganese-alkaline batteries, rated about 500 mAh. They are better in high drain use (such as in a radio) and might provide 100 mA for four or five hours.

Lithium-Ion batteries are more expensive, and are rated at 1,200 mAh, but are intended for moderate or low drain applications. They have internal fuses to prevent high discharge rates.

In very low drain applications — such as ionization-type smoke detectors or pacemakers — the battery life is limited by self-discharge rather than by external current drawn. Alkaline batteries might last six years, and some lithium batteries are designed to last 10 years or more — obviously, it is not desirable to open up a patient to replace pacemaker batteries often. That said, it's often advised to replace smoke detector batteries every six months. Use the old one in a radio, as it may still have some life left, but change them for safety.

Bart Bresnik
via email

ELECTRONET

INVEST in your BOT!



HiTEC

12115 Paine Street • Poway, CA 92064 • 858-748-6948 • www.hitecrd.com

USB Add USB to your next project--
It's easier than you might think! **DLP Design**

USB-FIFO • USB-UART • USB/Microcontroller Boards
RFID Readers • Design/Manufacturing Services Available
Absolutely NO driver software development required!
www.dlpdesign.com

PCB, PCBA and More! **Myro** Low cost High Quality

www.myropcb.com
1-888-PCB-MYRO

ALL ELECTRONICS CORPORATION
Electronic Parts and Supplies.
www.allelectronics.com
Free 96 page catalog 1-800-826-5432

For the ElectroNet online, go to
www.nutsvolts.com
click **Electro-Net**

superbrightleds.com
COMPONENT LEDs • LED BULBS • LED ACCENT LIGHTS



ADvertiser INDEX

■ LOOK FOR ■ SEARCH FOR ■ FIND

Find your favorite advertisers here!

AMATEUR RADIO / TV

Command Productions17
Palstar39

BUYING ELECTRONIC SURPLUS

All Electronics Corp.52

CIRCUIT BOARDS

Accutrace3
AP Circuits49
ExpressPCB35
Front Panel Express LLC49
Saelig Co., Inc.2

COMPONENTS

All Electronics Corp.52
Electronix Express43
Saelig Co., Inc.2
SDP/SI21

DESIGN/ENGINEERING/REPAIR SERVICES

Accutrace3
AP Circuits49
Cleveland Institute of Electronics34
ExpressPCB35
Front Panel Express LLC49

DEVELOPMENT PLATFORMS/TOOLS

AnarenBack Cover
M.E. Labs53

Technologic Systems15

EDUCATION

Cleveland Institute of Electronics34
Command Productions17
M.E. Labs53
PoLabs35

EMBEDDED SYSTEMS

Saelig Co. Inc.2
Technologic Systems15

EVENTS

CES66

ENCLOSURES

Front Panel Express LLC49

LCDs/DISPLAYS

EarthLCD21
Saelig Co., Inc.2

MICROCONTROLLERS / I/O BOARDS

M.E. Labs53
Technologic Systems15

MISC./SURPLUS

All Electronics Corp.52

MOTORS / MOTOR CONTROL

Robot Power21

SDP/SI21

ROBOTICS

Cleveland Institute of Electronics34
Electronix Express43
Robot Power21
SDP/SI21

TEST EQUIPMENT

Electronix Express43
Pico Technology67
PoLabs35
Saelig Co., Inc.2

TOOLS

PanaVise34
PoLabs35

WIRE, CABLE AND CONNECTORS

All Electronics Corp.52

WIRELESS PRODUCTS

AnarenBack Cover
Technologic Systems15

Accutrace3

All Electronics Corp.52

AnarenBack Cover

AP Circuits49

CES66

Cleveland Institute of

Electronics34

Command Productions17

EarthLCD21

Electronix Express43

ExpressPCB35

Front Panel Express LLC49

M.E. Labs53

Palstar39

PanaVise34

Pico Technology67

PoLabs35

Robot Power21

Saelig Co., Inc.2

SDP/SI21

Technologic Systems15



wha.

Innovation begins with a single word.



Consumer
Technology
Association™

Jan. 5-8, 2017 | Las Vegas, NV

Register now at CES.tech

#CES2017

Celebrating
50 YEARS
CES | 1967-2017

NEW

PicoScope[®] 2000 Series PC OSCILLOSCOPES

LIKE A BENCHTOP OSCILLOSCOPE,
ONLY SMALLER AND BETTER!

- 2 channel, 4 channel and MSO models
- Up to 100 MHz bandwidth
- Up to 128 MS buffer memory
- Function generator and AWG
- Decode 16 serial protocols
- USB-connected and powered
- Windows, Linux and Mac software
- Prices from \$129 to \$1,235



REDEFINING THE EXPECTATION OF PC SCOPES

All models include full software and 5 year warranty. Software includes measurements, spectrum analyzer, advanced triggers, color persistence, serial decoding (15 protocols including 1-Wire, CAN, Ethernet, I²C, I²S, LIN, RS-232, SENT, SPI, USB 1.1), masks, math channels, all as standard, with FREE updates. Free Software Development Kit also available.



Learn more about this new range at www.picotech.com
or give us a call at 1-800-591-2796 to discuss your application

"Fetch more coffee, Rex!"



A giant, coffee-carrying IoT dinobot at the command of your index finger? Why not. In fact, wherever your imagination leads you, Anaren empowers you with the industry's easiest-to-use wireless connectivity platform. Scan the code to learn more and register for a chance to win a free Atmosphere MSDK.* Think. Build. Connect.



Anaren®
What'll we think of next?®

* Must be 18 to participate. No purchase necessary. North America only. Void where prohibited. Other restrictions apply.

www.anaren.com/air/win